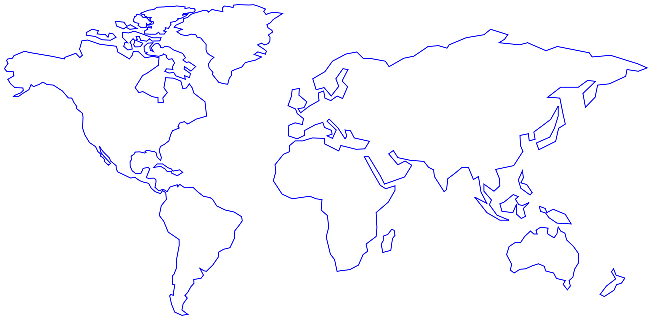


Common Criteria
for Information Technology
Security Evaluation

Trusted Product and Network Support
Evaluation Division

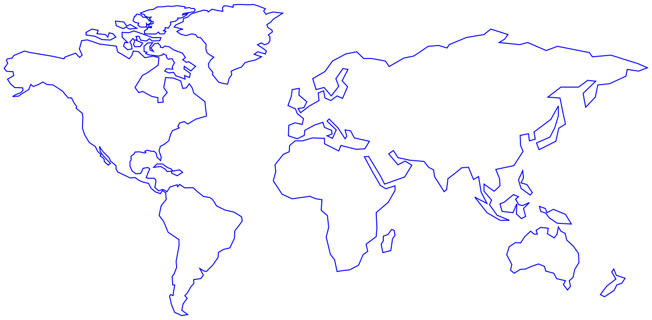
C2
Controlled Access
Protection Profile

This draft is still undergoing review and is subject to modification or withdrawl from publication. No reference to this document should be made in other publications.

National Security Agency
Ft. George G. Meade, MD 20755

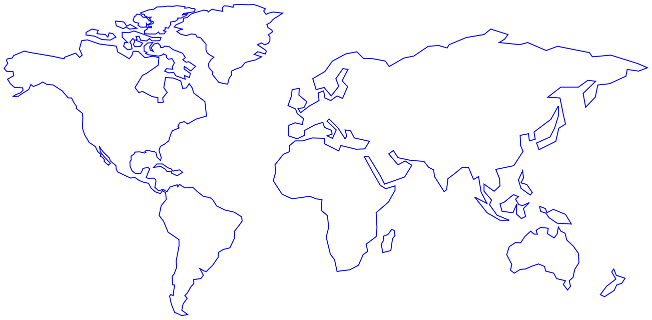
Version 2.0

August 21, 1997



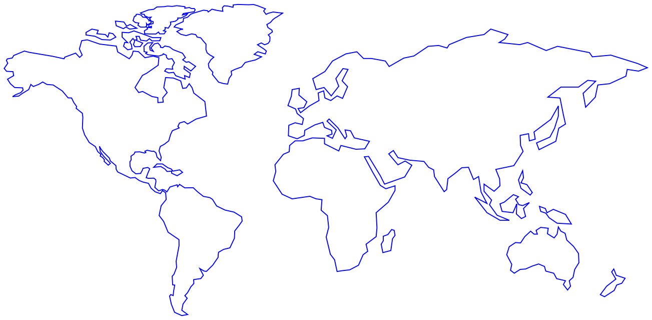
Common Criteria
for Information Technology
Security Evaluation

Version



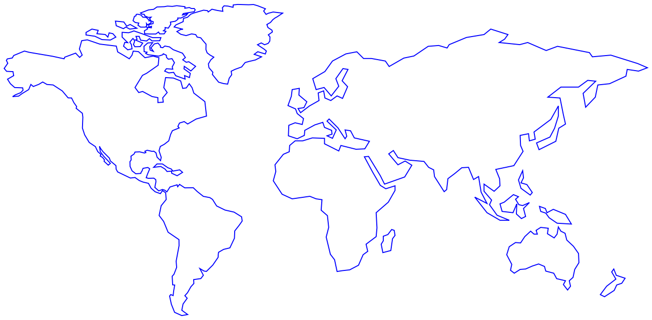
Common Criteria for Information Technology Security Evaluation

Version



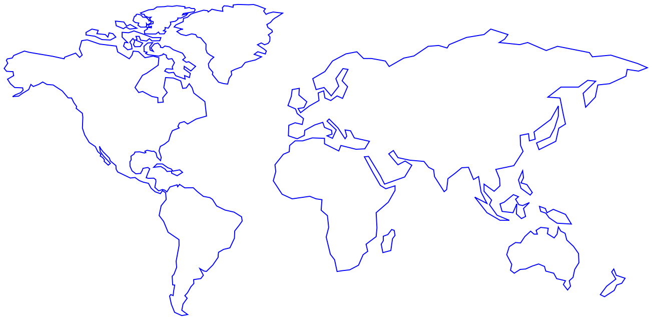
Common Criteria for Information Technology Security Evaluation

Version



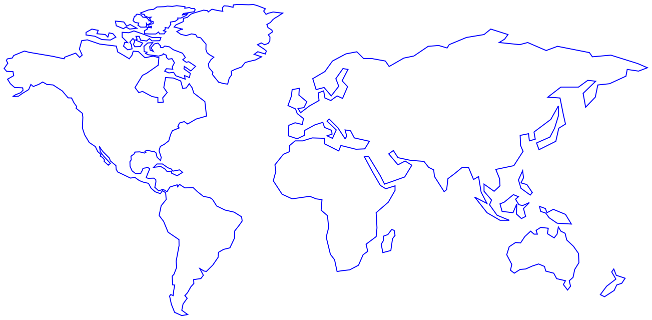
Common Criteria
for Information Technology
Security Evaluation

Version



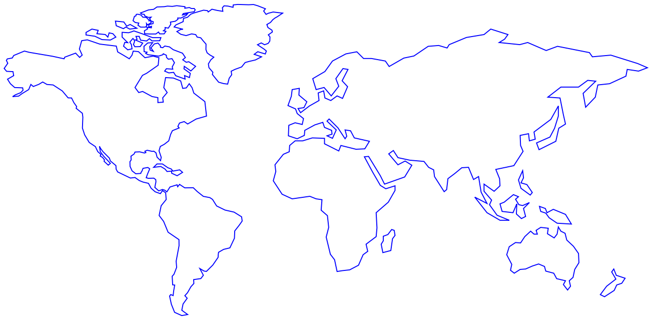
Common Criteria for Information Technology Security Evaluation

Version



Common Criteria for Information Technology Security Evaluation

Version



Common Criteria for Information Technology Security Evaluation

Version

Chapter 1

Introduction

This section contains document management and overview information necessary to allow a Protection Profile (PP) to be registered through a Protection Profile Registry. The PP *identification* provides the labelling and descriptive information necessary to identify, catalogue, register, and cross-reference a PP. The PP *overview* summarizes the profile in narrative form and provides sufficient information for a potential user to determine whether the PP is of interest. The overview can also be used as a stand-alone abstract for PP catalogues and registers.

1.1 Identification

Title: Controlled Access Protection (CAP).

Registration: TBD

Keywords: Access control, discretionary access control, general-purpose operating system, information protection

1.2 Protection Profile Overview

The Common Criteria (CC) CAP Protection Profile (PP), hereafter called CAP PP, specifies a set of security functional and assurance requirements for Information Technology (IT) products. CAP PP-conformant products support access controls that are capable of enforcing access limitations on individual users and data objects. CAP PP-conformant products also provide an audit capability which records the security-relevant events which occur within the system.

The CAP PP provides for a level of protection which is appropriate for an assumed non-hostile and well managed user community requiring protection against threats of inadvertent or casual attempts to breach the system security. The profile is not intended to be applicable to circumstances in which protection is required against determined attempts by hostile and well resourced attackers to breach system security. The CAP PP does not fully address the threats posed by system development or administrative personnel. CAP PP-conformant products are suitable for use in both commercial and government environments.

The CAP PP was derived from the requirements of the C2 class of the U.S. Department of Defense (DoD) *Trusted Computer System Evaluation Criteria (TCSEC)*, dated December, 1985, and the material upon which those requirements are based. This protection profile provides security functions and assurances which are nearly equivalent to those provided by the TCSEC and replaces the requirements used for C2 trusted product evaluations.

The CAP PP is generally applicable to distributed systems but does not address the security requirements which arise specifically out of the need to distribute the resources within a network.

1.3 Use of Terms

This profile uses the following terms which are described in this section to aid in the application of the requirements:

- *User*
- *Authorized User*
- *Authorized Administrator*
- *Discretionary Access Control (DAC) Policy*
- *Mediation*
- *Access*
- *Authorization*

A *user* is a named individual who attempts to invoke a service offered by the TOE.

There are two types of users referenced in this profile: authorized users and authorized administrators. *Authorized users* are those individuals who request a service from the TOE and have been authenticated but have no responsibility (or associated TOE privilege) to administer the TOE. *Authorized administrators* are individuals who have been authenticated by the TOE and have been assigned responsibility (and associated TOE privilege) to administer the TOE.

Whether a user is granted a requested action is determined by the TOE Security Policy (TSP) which is specified in this profile in the context of Discretionary Access Control (DAC). The *DAC policy* is the set of rules used to mediate user access to TOE protected objects and can be generally characterized as a policy which requires the TOE to allow authorized users and authorized administrators to control access to objects based on individual user identification. When the DAC policy rules are invoked, the TOE is said to be *mediating* access to TOE protected objects. However, there may be instances when the DAC policy is not invoked meaning that there may be objects residing in the TOE which are not protected by the TSP. In these instances the TOE is said to not be mediating access to a set of objects even though the TOE is executing a (possibly unauthorized) user request.

The DAC policy consists of two types of rules: those which apply to the behavior of authorized users (termed access rules) and those which apply to the behavior of authorized administrators (termed authorization rules). If an authorized user is granted a request to operate on an object, the user is said to have *access* to that object. There are numerous types of access; typical ones include read access and write access which allow the reading and writing of objects respectively. If an authorized administrator is granted a requested service, the user is said to have *authorization* to the requested service or object. Like access, there a numerous possibilities for authorizations. Typical authorizations include auditor authorization which allows an administrator to view audit records and execute audit tools and DAC override authorization which allows an administrator to override object access controls to administer the system.

Chapter 2

TOE Description

The CAP PP defines a set of security requirements to be levied on Targets of Evaluation (TOEs) which include workstations, mainframes, general-purpose operating systems (including the hardware platform). Such TOEs permit one or more processors and attached peripheral and storage devices to be used by multiple users to perform a variety of functions requiring controlled, shared access to the data stored on the system. Such installations are typical of personal, workgroup, or enterprise computing systems accessed by users local to, or with otherwise protected access to, the computer systems.

The CAP PP is applicable to TOEs that provide facilities for on-line interaction with users, as well as TOEs that provide for batch processing. The protection profile is also generally applicable to TOEs incorporating network functions but contains no network specific requirements. Networking is covered only to the extent to which the TOE can be considered to be part of a centrally-managed system that meets a common set of security requirements.

The CAP PP assumes that responsibility for the safeguarding of the data protected by the TOEs security functions (TSF) can be delegated to the TOE users. All data is under control of the TOE; however, not all objects are required to be under control of the TSF. The data are stored in objects, and the TSF can associate with each controlled object a description of the access rights to that object.

All individual users are assigned a unique identifier. This identifier supports individual accountability. The TSF authenticates the claimed identity of the user before allowing the user to perform any actions that require TSF mediation.

Chapter 3

Security Environment

3.1 Threats

The CAP PP has derived all security objectives from the statement of Organizational Security Policy found in the following section. Therefore, there is no statement of the explicit threats countered by the CAP PP.

3.2 Organizational Security Policies

An Organizational Security Policy is a set of rules or procedures imposed by an organization upon its operations to protect its sensitive data. Although the organizational security policies described below are drawn from DoD Manual 5200.28-M (Techniques and procedures for Implementing, Deactivating and Evaluating Resource Sharing ADP Systems) they apply to many non-DoD environments.

P.ACCESS_RULES

Given identified subjects and objects, there must be a set of rules that are used by the TOE to determine whether a given subject can be permitted to gain access to a specific object. As a default the protection must be restricted to the creator of the object, unless otherwise specified.

P.ACCESS_ENFORCE

Discretionary security controls are required to ensure that only selected authorized users or groups of users may obtain access to data (e.g., based on a need-to-know).

P.KNOWN

Legitimate users of the TOE must be identified before TOE access can be granted.

P.AUTH

Each access to information must be mediated based on who is accessing the information and information they are authorized to access.

P.AUTH_PROTECT

The identification and authorization information must be securely maintained by the TOE and be associated with every active element that performs some security-relevant action in the system.

P.ACCOUNT

The TOE must ensure that all TOE users can subsequently be held accountable for their security-relevant actions.

P.OPERATIONAL_ASSURE

Features must be available that can be used to periodically validate the correct operation of the on-site hardware and firmware elements of the TSF.

P.MANAGE

The TOE is managed by authorized users.

3.3 Security Usage Assumptions

This section describes the security aspects of the environment in which the TOE will be, or is intended to be used. This includes information about the physical, personnel, and connectivity aspects of the environment.

A CAP PP-conformant TOE is assured to provide effective security measures in a cooperative non-hostile environment only if it is installed, managed, and used correctly. The operational environment must be managed in accordance with assurance requirements documentation for delivery, operation, and user/ administrator guidance. The following specific conditions are assumed to exist in an environment where CAP PP-conformant TOEs are employed:

3.3.1 Physical Assumptions

CAP PP-conformant TOEs are intended for application in user areas that have physical control and monitoring. It is assumed that the following physical conditions will exist:

A.LOCATE

The processing resources of the product, excluding dumb terminals, will be located within controlled access facilities which will prevent unauthorized physical access.

A.PROTECT

The TOE hardware and software critical to security policy enforcement will be physically protected from unauthorized modification.

3.3.2 Personnel Assumptions

It is assumed that the following personnel conditions will exist:

A.MANAGE

There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.

A.NO_EVIL_ADM

The system administrative personnel are not careless, wilfully negligent, or hostile, and will follow and abide by the instructions provided by the administrator documentation.

A.COOP

Users possess the necessary authorization to access at least some of the information managed by the TOE and are expected to act in a cooperating manner in a benign environment.

3.3.3 Connectivity Assumptions

The CAP PP contains no explicit network or distributed system requirements. However, it is assumed that the following connectivity conditions exist:

A.PEER

Any other systems with which the TOE communicates are assumed to be under the same management control and operate under the same security policy constraints.

CAP PP-conformant TOEs are applicable to networked or distributed environments only if the entire network operates under the same constraints and resides within a single management domain. There are no security requirements which address the need to trust external systems or the communications links to such systems.

A.CONNECT

All connections to peripheral devices reside within the controlled access facilities.

CAP PP-conformant TOEs only address security concerns related to the manipulation of the TOE through its legitimate interfaces. Internal communication paths to interface points such as terminals are assumed to be adequately protected.

Chapter 4

Security Objectives

This section defines the security objectives of the TOE and its supporting environment. Security objectives, categorized as either IT security objectives or non-IT security objectives, reflect the stated intent to counter identified threats and/or comply with any organizational security policies identified. All of the identified threats and organizational policies are addressed under one of the categories below.

4.1 IT Security Objectives

The following are the CAP PP IT security objectives:

O.ACCESS

The TOE must ensure that only authorized users gain access to the TOE and its resources.

O.CONTROLLED_ACCESS

The TOE must allow users to specify access control and sharing of objects based on identified individuals or groups of individuals.

O.DAC_ENFORCE

The TOE must include provisions for the enforcement of discretionary access control rules.

O.ACCOUNT

The TOE must provide the capability to selectively keep and protect audit information associated with individual users.

O.ACCT_MANAGE

The TOE must provide the capability for an authorized user to access and evaluate accountability information by a secure means.

O.OPERATIONAL_ASSURE

Feature(s) must be provided that allow a site to periodically validate the correct operation of the TSF's hardware and firmware.

O.SYS_ARCH

The TOE must provide a defined set of resources that the TSF controls.

O.ISOLATE

The TSF must maintain a domain for its execution that protects itself from external interference or tampering.

O.REUSE

The TOE must ensure that residual information is made unavailable for unauthorized reuse.

O.MANAGE

The TOE must provide all the functions and facilities necessary to support the authorized users that are responsible for the management of TOE security.

O.AUTH

Trusted users have associated authorization(s) that allow access to data not intended to be accessed by untrusted users.

4.2 Non-IT Security Objectives

A CAP PP-conformant TOE is assumed to be complete and self-contained and, as such, is not dependent upon any other products to perform properly. However, certain objectives with respect to the general operating environment must be met in order to support the C2 security capabilities. The following are the CAP PP non-IT security objectives:

O.INSTALL

Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner which maintains IT security.

O.PHYSICAL

Those responsible for the TOE must ensure that those parts of the TOE critical to security policy are protected from physical attack which might compromise IT security.

O.CREDEN

Those responsible for the TOE must ensure that all access credentials are protected by the users in a manner which maintains IT security.

Chapter 5

Security Requirements

This chapter lists the IT security requirements that must be satisfied by a CAP PP-conformant TOE in order to meet its security objectives. These requirements consist of functional components from Part 2 of the CC and an Evaluation Assurance Level (EAL) containing assurance components from Part 3 of the CC. The functional and assurance requirements are listed in Tables 5.1 and 5.2, respectively. The actual text of these requirements is provided in Chapter 6.

5.1 Functional Requirements

This section defines the functional requirements for the TOE. All functional requirements components in this profile were drawn from Part 2 of the CC.

CC defined operations for assignment, selection, and refinement were used to tailor the requirements to the level of detail necessary to meet the stated security objectives in Section 4.1. These operations are indicated through the use of italicized text. The use of these operations does not constrain TOE implementation, and all required operations not performed within this profile are clearly identified and described such that they can be correctly performed upon instantiation of the PP into a Security Target (ST) specification.

Table 5.1 lists the CAP PP functional components. The remainder of this section is organized as shown in the table. There is one subsection for each class of functional components included in the profile. The contents of each subsection consists of the text of the components selected.

	Component	Name	Refined	CCOR
Security Audit Class				
1	FAU_GEN.1	Audit Data Generation		
2	FAU_GEN.2	User Identity Generation		
3	FAU_MGT.1	Audit Trail Management		
4	FAU_MGT.2	Audit Trail Saturation Control		
5	FAU_PRO.2	Extended Audit Trail Access		
6	FAU_SAR.2	Extended Audit Review		
7	FAU_SAR.3	Selectable Audit Review	X	
8	FAU_SEL.1	Selective Audit		
9	FAU_STG.3	Prevention of Audit Data Loss		
User Data Protection Class				
10	FDP_ACC.1	Subset Object Access Control		
11	FDP_ACF.1	Single Security Attribute Access Control		
12	FDP_ACI.1	Static Attribute Initialisation		
13	FDP_RIP.3	Full Residual Information Proteciton on Allocation		
14	FDP_SAM.2	User Attribute Modification		
Identification and Authentication Class				
15	FIA_ADA.3	Expanded User Authentication Data Administration		
16	FIA_ADP.1	Basic User Authentication Data Protection		
17	FIA_ADP.2	Extended User Authentication Data Protection		
18	FIA_ATA.3	Extended User Attribute Administration		
19	FIA_ATD.1	User Attribute Definition		
20	FIA_UAU.8	Timing of Authentication	X	X
21	FIA_UID.2	User Identification		X
22	FIA_UID.3	Timing of Identification	X	
23	FIA_USB.1	User-Subject Binding		
Protection of the Trusted Security Functions Class				
24	FPT_AMT.1	Abstract Machine Testing		
25	FPT_REV.1	Basic Revocation		
26	FPT_REV.2	Immediate Revocation	X	
27	FPT_RVM.1	Non-Bypassability of TSP		
28	FPT_SEP.1	TSF Domain Separation		
29	FPT_TSA.1	Basic Security Administration		

Table 5.1 - Functional Components of the CAP PP

5.1.1 Security Audit Components

FAU_GEN.1: Audit Data Generation

FAU_GEN.1.1: The TSF shall be able to generate an audit record of the following auditable events:

a) Startup and shutdown of the audit functions.

b) All auditable events for the *basic* level of audit, as defined in all functional components included in the PP/ST; and

c) Based on all functional components included in the PP/ST,[1]

- *The use of security authorizations (e.g., privileges) to override enforcement of the SFP;*
- *All actions by administrators;*
- *Establishment of a path that connects a user to another users process;*
- *Successful and unsuccessful attempts to specify the granting or denying of access to an object; and*
- *[assignment: other auditable events].*

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of event, type of event, subject identity, and *success or failure* of the event.

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST,

- *For Identification & Authentication events, the origin (e.g., terminal ID) of the request;*
- *For introduction of an object into a user's address space, and for object deletion, the object identity;*
- *For administrator changes to security-relevant databases, the new values for changed items;*
- *For administrator actions in a TOE with multiple operator consoles, the identity of the console from which the auditable event originated; and*
- *[assignment: other audit relevant information].*

FAU_GEN.2: User Identity Generation

FAU_GEN.2.1: The TSF shall be able to associate any auditable event with the identity of the user that caused the event.

FAU_MGT.1: Audit Trail Management

[1] These are additional auditable events not covered in a) or b) above.

FAU_MGT.1.1: The TSF shall provide the authorized administrator with the ability to *create, delete, and empty* the audit trail.

FAU_MGT.2: Audit Trail Saturation Control

FAU_MGT.2.1: The TSF shall generate an alarm to the authorized administrator if the size of the audit data in the audit trail exceeds a *[assignment: pre-defined limit]*.

FAU_PRO.2: Extended Audit Trail Access

FAU_PRO.2.1: The TSF shall restrict full access to the audit trail to the authorized administrator.

FAU_PRO.2.2: The TSF shall provide only authorized users with the capability to read *[assignment: list of audit information]* from the audit trail.

FAU_SAR.2: Extended Audit Review

FAU_SAR.2.1: The TSF shall provide audit review tools, with the ability to view the audit data.

FAU_SAR.2.2: The TSF shall restrict full use of the audit review tools to the authorized administrator.

FAU_SAR.3: Selectable Audit Review

FAU_SAR.3.1:[2] The TSF shall provide audit review tools with the ability to perform *[selection: searches, sorting]*[3] of audit data based on *user identity and [assignment: other logical operations] if audit collection data cannot be restricted on the basis of user identity*.[4]

FAU_SEL.1: Selective Audit

FAU_SEL.1.1: The TSF shall be able to include or exclude auditable events from the set of audited events based on *one or more of* the following attributes:

a) *User identity; and*

b) *[assignment: list of additional attributes]* that audit selectivity is based upon.

if the audit review tools cannot search or sort data on the basis of user identity[5].

FAU_STG.3: Prevention of Audit Data Loss

[2] A CCOR concerning a slight wording change is currently being processed. [PPTeam33]
[3] A CCOR concerning this change has been submitted to the CC [c2pp97.5].
[4] A CCOR concerning this change has been submitted to the CC [c2pp97.10].
[5] A CCOR concerning this change has been submitted to the CC [c2pp97.10].

FAU_STG.3.1: The TSF shall store generated audit records in a permanent audit trail.

FAU_STG.3.2: The TSF shall limit the number of audit records lost due to system *audit storage exhaustion and failure*.

FAU_STG.3.3: In the event of audit storage exhaustion, the TSF shall be capable of *preventing* the occurrence of auditable actions, except those taken by the authorized administrator.

5.1.2 User Data Protection Components

FDP_ACC.1: Subset Object Access Control

FDP_ACC.1.1: The TSF shall enforce the *Discretionary Access Control Policy* on *named individuals, subjects acting on behalf of named individuals, groups of named individuals, and [assignment: list of named objects]* for *[assignment: operations among subjects and named objects covered by the Discretionary Access Control Policy]*.

FDP_ACF.1: Single Security Attribute Access Control

FDP_ACF.1.1: The TSF shall enforce the *Discretionary Access Control Policy* to objects based on *the following subject attributes:*

a) *named individuals;*

b) *defined groups of individuals;*

c) *[assignment: list of subject authorizations].*

FDP_ACF.1.1: The TSF shall enforce the *Discretionary Access Control Policy* to objects based on *the following named object attributes:*

a) *access rights.*

FDP_ACF.1.2: The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

a) *[assignment: rules governing access among subjects and named objects using operations on named objects].*

FDP_ACI.1: Static Attribute Initialization

FDP_ACI.1.1: The TSF shall enforce the *Discretionary Access Control Policy* to provide *restrictive* default values for the object security attributes that are used to enforce the *Discretionary Access Control Policy*.

FDP_ACI.1.2: The TSF shall allow the specification of alternate initial values to override the default values when the object is created.

FDP_RIP.3: Full Residual Information Protection on Allocation

FDP_RIP.3.1: The TSF shall ensure that upon the allocation of a resource to all objects any previous information content is made unavailable.

FDP_SAM.2: User Attribute Modification

FDP_SAM.2.1: The TSF shall enforce *the Discretionary Access Control Policy* to provide authorized users with the ability to modify:

a) *named object access control specifications, based on named individual and/or defined groups of named individuals.*

b) *[assignment: other security attributes]*

5.1.3 Identification and Authentication Components

FIA_ADA.3: Expanded User Authentication Data Administration

FIA_ADA.3.1: The TSF shall provide functions for initializing and modifying user authentication data related to *[assignment: identified authentication mechanism]*.

FIA_ADA.3.2: The TSF shall restrict the use of these functions on the user authentication data for any user to the authorized administrator.

FIA_ADA.3.3: The TSF shall allow authorized users to use these functions to modify their own authentication data in accordance with the TSP.

FIA_ADP.1: Basic User Authentication Data Protection

FIA_ADP.1.1: The TSF shall protect from unauthorized observation, modification, and destruction authentication data that is stored in the TOE.

FIA_ADP.2: Extended User Authentication Data Protection

FIA_ADP.2.1: The TSF shall protect from unauthorized observation, modification, and destruction the raw form of authentication data at all times while it resides in the TOE.

FIA_ATA.3 Extended User Attribute Administration

FIA_ATA.3.1 The TSF shall provide the ability to *display and modify* user attributes.

FIA_ATA.3.2 The TSF shall limit the ability to modify any user's attributes to only the authorised administrator.

FIA_ATA.3.3 The TSF shall allow users to modify their own attributes in accordance with the TSP.

FIA_ATD.1: User Attribute Definition[6]

FIA_ATD.1: The TSF shall provide, for each user, a set of security attributes necessary to enforce the TSP.

FIA_UAU.8: Timing of Authentication[7]

FIA_UAU.8.1: The TSF shall allow users to perform *no actions requiring TSF mediation* before the user's claimed identity is authenticated.

[6] An FRG Note was written about whether to select FIA_ATD.1 or FIA_ATD.2. Since the FRG recommends using FIA_ATD.1, there is no issue here. [FRG Note 7]

[7] An FRG Note was written about the assignment statements for this component. However, despite the FRG recommendation to change the assignment statement, nothing was changed from the original CC [FRG Note 8]. A CCOR was written [PPTeam29] stating that there was an incorrect dependency. The team received a favourable FRG response.

FIA_UAU.8.2: The TSF shall perform the authentication of any user's claimed identity prior to performing any TSF-mediated actions on behalf of the user.

FIA_UID.2: User Identification

FIA_UID.2.1: The TSF shall uniquely identify each user.[8]

FIA_UID.3: Timing of Identification

FIA_UID.3.1: The TSF shall allow users to perform *only actions that do not require access to the TOE information, services, or resources that are restricted by policy on the basis of user identity or other security attributes* before identifying the user.

FIA_UID.3.2: The TSF shall identify each user before performing any other actions on behalf of the user.

FIA-USB.1: User-Subject Binding

FIA_USB.1.1: The TSF shall associate the appropriate user security attributes with subjects acting on behalf of that user.

[8] A CCOR has been submitted for this change [c2pp97.7].

5.1.4 Protection of the Trusted Security Functions Components

FPT_AMT.1: Abstract Machine Testing

FPT.AMT.1.1: The TSF shall provide the authorized administrator with the capability to demonstrate the correct operation of the security-relevant functions provided by the TSF's underlying abstract machine.

FPT_REV.1: Basic Revocation

FPT_REV.1.1: The TSF shall provide a capability for revocation of security attributes associated with the *named users, named objects and [assignment: list of additional resources]* within the TSC[9].

FPT_REV.1.2: The TSF shall enforce revocation *of:*

a) *Discretionary Access Control security attributes. Changes shall have an effect for new access requests.*

b) *[assignment: specification of other revocation rules].*

FPT_REV.2: Immediate Revocation

FPT_REV.2.1: The TSF shall provide a capability for revocation of security attributes associated with the *named users and [assignment: list of additional resources]* within the TSC .

FPT_REV.2.2: The TSF shall immediately enforce revocation of *security-relevant authorizations.*

FPT_RVM.1: Non-Bypassability of TSP

FPT_RVM.1.1: The TSF shall ensure that the TSP enforcement functions are invoked and succeed before any security-related operation is allowed to proceed.

FPT_SEP.1: TSF Domain Separation

FPT_SEP.1.1: The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2: The TSF shall enforce separation between the security domains of subjects in the TSC.

FPT_TSA.1: Basic Security Administration

FPT_TSA.1.1: The TSF shall distinguish security-relevant administrative functions from other functions.

[9] TOE Scope of Control

FPT_TSA.1.2: The TSF's set of security-relevant administrative functions shall include all functions necessary to install, configure, and manage the TSF; minimally, this set shall include: *[assignment: list of administrative services to be minimally supplied].*

FPT_TSA.1.3: The TSF shall restrict the ability to perform security-relevant administrative functions to specifically authorized users.

FPT_TSA.1.4: The TSF shall be capable of distinguishing the set of users authorized for administrative functions from the set of all users of the TOE.

5.2 Assurance Requirements

This section defines the assurance requirements for the TOE using assurance requirements components drawn from Part 3 of the CC. Any required operations are used to amplify the requirements to the level of detail necessary so that the security objectives are met.

Refinement operations that have been carried out in the profile are indicated through the use of italicized text.

Table 5.2 lists the CAP PP assurance components. The remainder of this section is organized as shown in the table. There is one subsection for each class of assurance components included in the profile. The contents of each subsection consists of the text of the components selected for the profile.

	Component	Name	Refined	CCOR
Configuration Management Class				
1	ACM_CAP.2	Authorization Controls		
2	ACM_SCP.1	Minimal CM Coverage		
Delivery and Operation Class				
3	ADO_IGS.1	Installation, Generation, and Start-up Procedures		
Development Class				
4	ADV_FSP.1	TOE and Security Policy	X	X
5	ADV_HLD.2	Security Enforcing High-Level Design	X	XX
6	ADV_RCR.1	Informal Correspondence Demonstration		
Guidance Documents Class				
7	AGD_ADM.1	Administrator Guidance	X	X
8	AGD_USR.1	User Guidance	X	X
Life-Cycle Support Class				
9	ALC_DVS	Identification of Security Measures		
Tests Class				
10	ATE_COV.2	Complete Coverage - Rigorous		
11	ATE_DPT.2	Testing - High-Level Design		
12	ATE_FUN.1	Functional Testing		
13	ATE_IND.2	Independent Testing - Sample		
Vulnerability Assessment Class				
14	AVA_MSU.1	Misuse		
15	AVA_SOF.1	Strength of TOE Security Function Evaluation		
16	AVA_VLA.1	Developer Vulnerability Analysis		X

Table 5.2 - Assurance Components of the CAP PP

5.2.1 Configuration Management (CM) Components

ACM_CAP.2: Authorization Controls

ACM_CAP.2.1D**:** The developer shall use a CM system.

ACM_CAP.2.2D**:** The developer shall provide CM documentation.

ACM_CAP.2.1C**:** The CM documentation shall include a configuration list and a CM plan.

ACM_CAP.2.2C**:** The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.2.3C**:** The CM documentation shall describe the method used to uniquely identify the TOE configuration items.

ACM_CAP.2.4C**:** The CM plan shall describe how the CM system is used.

ACM_CAP.2.5C**:** The CM documentation shall provide evidence that the CM system is working properly.

ACM_CAP.2.6C**:** The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.

ACM_CAP.2.7C**:** The CM system shall ensure that only authorized changes are made to the TOE configuration items.

ACM_CAP.2.1E**:** The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ACM_SCP.1: Minimal CM Coverage

ACM_SCP.1.1D: The developer shall provide CM documentation.

ACM_SCP.1.1C: As a minimum, the following shall be tracked by the CM system: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, and CM documentation.

ACM_SCP.1.2C: The CM documentation shall describe how configuration items are tracked by the CM system.

ACM_SCP.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.2 Delivery and Operation Components

ADO_IGS.1: Installation, Generation, and Start-up Procedures

ADO_IGS.1.1D: The developer shall document procedures to be used for the secure installation, generation, and start-up of the TOE.

ADO_IGS.1.1C: The documentation shall describe the steps necessary for secure installation, generation, and start-up of the TOE.

ADO_IGS.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.3 Development Components

ADV_FSP.1: TOE and Security Policy

ADV_FSP.1.1D: The developer shall provide a functional specification.

ADV_FSP.1.2D: The developer shall provide a TSP.

ADV_FSP.1.1C: The functional specification shall describe the TSF using an informal style.

ADV_FSP.1.2C: The functional specification shall include an informal presentation of syntax and semantics of all external TSF interfaces.

ADV_FSP.1.3C: The functional specification shall include evidence that demonstrates that the TSF is completely represented.

ADV_FSP.1.4C: *The functional specification shall include the following information:*[10]

a) *[FAU_GEN.2]* A specification of the manner in which an auditable event is associated with the identity of a user.[11]

b) *[FIA_ADP.1]* A specification of the manner by which the user authentication data is protected from unauthorized use, observation, modification and destruction while the authentication data is stored in the TSF.

c) *[FIA_ADP.2]* A specification of the manner by which the user authentication data is protected from unauthorized use, observation, modification and destruction at all times while it is under TSF control.

d) *[FPT_TSA]* A specification of the security-relevant administrative functions in the TSF.

e) *[FIA_ADA]* A specification of the TSF authentication data administration mechanism.

f) *[FIA_ATD]* A specification of the user-related TSP attributes and the manner in which they are associated with the user.

g) *[FIA_UAU]* A specification of the TSF authentication mechanism(s).

h) *[FIA_UID]* A specification of the user identification function.

[10] This is a new element not contained in the CC. A CCOR has been written and submitted to the FRG [PPTeam1.0]. This element is a result of the FRG's recommendation.

[11] A CCOR concerning this element is currently being reviewed. [PPTeam51]

i) *[FIA_USB]* A specification of the manner in which user attributes are associated with subjects that the user owns.

ADV_FSP.1.5C: *[FDP_ACC.1] The functional specification must define the subset of operations and objects controlled by the policy, describe the intended use of these operations, and provide a detailed rationale for the scope of the subset* .[12]

ADV_FSP.1.6C: *[FDP_ACC.1] The functional specification should be used to provide evidence for the rationale that the objects are covered by the access control Security Function Policies (SFPs) and that there are no conflicts in the case of multiple SFPs (See footnote).*

ADV_FSP.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.1.2E: The evaluator shall determine that the functional specification is consistent with the TSP.

ADV_FSP.1.3E: The evaluator shall determine if the functional requirements in the ST are addressed by the representation of the TSF.

ADV_HLD.2: Security Enforcing High-level Design

ADV_HLD.2.1D: The developer shall provide the high-level design of the TSF.

ADV_HLD.2.1C: The presentation of the high-level design shall be informal.

ADV_HLD.2.2C: The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.2.3C: The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.2.4C: *The high-level design shall describe the external interfaces of the TSF*.[13]

ADV_HLD.2.5C: The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV_HLD.2.6C: The high-level design shall describe the separation of the TSF into TSP enforcing and other subsystems.

ADV_HLD.2.7C: *The high-level design shall include the following information:*[14]

[12] A CCOR concerning this element is currently being reviewed [PPTeam1.1].
[13] This modifcation to the CC has been submitted as a CCOR [c2pp97.1].
[14] This is a new element not contained in the CC. A CCOR has been written and sumitted to the FRG [PPTeam1.0]. This element is a result of the FRG's recommendation.

a) *[FIA_UID]* A specification of the user identification function.

b) *[FPT_SEP]* A description of the architecture and design of the domain separation mechanism.

ADV_HLD.2.8C: *The high-level design shall describe how the correct operation of the security-relevant functions provided by the TSF's underlying abstract machine is demonstrated.*[15]

ADV_HLD.2.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_HLD.2.2E: The evaluator shall determine if the functional requirements in the ST are addressed by the representation of the TSF.

ADV_RCR.1: Informal Correspondence Demonstration

ADV_RCR.1.1D: The developer shall provide evidence that the least abstract TSF representation provided is an accurate, consistent, and complete instantiation of the functional requirements expressed in the ST.

ADV_RCR.1.1C: For each adjacent pair of TSF representations, the evidence shall demonstrate that all parts of the more abstract representation are refined in the less abstract representation.

ADV_RCR.1.2C: For each adjacent pair of TSF representations, the demonstration of correspondence between the representations may be informal.

ADV_RCR.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_RCR.1.2E: The evaluator shall analyze the correspondence between the functional requirements expressed in the ST and the least abstract representation provided to ensure accuracy, consistency, and completeness.

[15] A CCOR concerning this new element has been written [c2pp97.9].

5.2.4 Guidance Documents Components

AGD_ADM.1: Administrator Guidance

AGD_ADM.1.1D: The developer shall provide administrator guidance addressed to system administrative personnel.

AGD_ADM.1.1C: The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD_ADM.1.2C: The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_ADM.1.3C: The administrator guidance shall contain guidelines on the consistent and effective use of the security functions within the TSF.

AGD_ADM.1.4C: The administrator guidance shall describe the difference between two types of functions: those which allow an administrator to control security parameters, and those which allow the administrator to obtain information only.

AGD_ADM.1.5C: The administrator guidance shall describe all security parameters under the administrator's control.

AGD_ADM.1.6C: The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_ADM.1.7C: The administrator guidance shall contain guidelines on how the security functions interact.

AGD_ADM.1.8C: The administrator guidance shall contain instructions regarding how to configure the TOE.

AGD_ADM.1.9C: The administrator guidance shall describe all configuration options that may be used during secure installation of the TOE.

AGD_ADM.1.10C: The administrator guidance shall describe details, sufficient for use, of procedures relevant to the administration of security.

AGD_ADM.1.11C: The administrator guidance shall be consistent with all other documents supplied for evaluation.

AGD_ADM.1.12C: *[FAU_GEN] The administrator guidance shall provide a description of the audit record format.*[16]

AGD_ADM.1.13C: *[FAU_GEN] The administrator guidance shall provide a list of auditable events* [17].

AGD_ADM.1.14C: *The administrator guidance shall contain guidance on or a description of:* [18]

a) *[FAU_MGT]* Recommendations for handling notifications generated by the TSF when a security violation appears imminent[19].

b) *[FAU_MGT]* Recommendations for defining the limit to control the audit trail saturation[20].

c) *[FAU_SAR]* Using the audit review tools.[21]

d) *[FDP_ACI.1]* Identify acceptable alternate initial values for object security attributes if the default values are over-ridden.[22]

e) *[FDP_SAM.1]* How security attributes associated with objects, users, or subjects are modified.[23]

f) *[FIA_ADA]* Using the TSF authentication data administration mechanism.

g) *[FIA_UAU]* Configuring the TSF authentication mechanism(s).

h) *[FIA_UAU.8]* Explaining the risks in placing passwords on card input and suggesting procedures to mitigate that risk.

i) *[FIA_UID]* How to define users.

j) *[FPT_AMT]* Using the product features that can be used to periodically demonstrate the correct operation of the underlying abstract machine

k) *[FPT_AMT]* The coverage and use of the underlying abstract machine tests.

l) *[FPT_REV.1]* The timing aspects of the revocation.

m) *[FPT_TSA]* The initial configuration of the security-relevant administrative commands and (if applicable) the roles with which they are associated.

n) *[FPT_TSA]* The TSF facilities used by an authorized administrator to define security-relevant administrative commands and (if applicable) associate them with a role.

[16] A CCOR has been submitted to the FRG [PPTeam1.0] and an FRG response has been forwarded to the team. However, the recommendation was to create a new element, AGD_ADM.1.12C which is currently AGD_ADM.1.14C in this profile**.** A separate CCOR has been written to address the use of “should” and “shall” within the documentation notes [PPTeam1.1]. This element was introduced to the profile as a result of pulling a documentation note that used “should” and making it a required element.

[17] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.

[18] This is a new element not contained in the CC. A CCOR has been written and sumitted to the FRG [PPTeam1.0]. This element is a result of the FRG’s recommendation.

o) *[FPT_TSA]* The responsibilities of the security-relevant administrative role(s), as applicable.

p) *[FAU_MGT]* Recommendations for handling notifications generated by the TSF when a security violation appears imminent[24].

q) *[FAU_MGT]* Recommendations for defining the limit to control the audit trail saturation[25].

r) *[FAU_PRO]* A description of the protection rules for the audit trail *(See footnote)*.

s) *[FAU_PRO]* An identification of the rules for managing access to the audit trail by users[26].

t) *[FAU_SEL]* A description of the selection rules for the audit events *(See footnote)*.

u) *[FAU_SEL]* An identification of the rules for managing the auditable set of events[27].

v) *[FAU_STG]* An identification of the conditions under which loss of audit data due to system failure shall be enumerated and the potential number of audit events lost shall be documented[28].

w) *[FDP_ACC]* Guidance with respect to each access control policy satisfying a FDP_ACC component. Documentation shall be provided for end-users, authorized administrative users, or both, as appropriate for the nature of the objects and operations controlled by the policy[29].

x) *[FDP_ACC.1]* A definition of the subset of operations controlled by the SFP, describe the intended use of the operations, and provide a detailed rationale for the scope of the subset. The rationale shall be sufficient to convince the evaluator that all listed objects are covered by the access control SFP. In the case of multiple SFPs, rationale should be provided to demonstrate that the SFPs do not conflict. If the PP/ ST author claims that there is complete coverage of all objects and operations within the TOE Scope of Control (TSC), then rationale shall be provided to demonstrate this as well as that no conflicts exist between the SFPs[30].

y) *[FDP_ACF]* Information detailing what is the basis of mediation, what is the precedence of mediation when more than one conclusion could be reached given a set of attributes, etc.[31].

[19] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.

[20] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.

[21] A CCOR concerning this element is currently being reviewed. [PPTeam50]

[22] A CCOR concerning this element is currently being reviewed. [PPTeam52]

[23] A CCOR concerning this element is currently being reviewed. [PPTeam53]

[24] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.

[25] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.

[26] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.

z) *[FDP_ACF]* A description of the nature and scope of each access control policy and briefly describe the functions that implement the policy (FDP_ACF), the security attributes that govern the policy (FDP_SAQ, FDP_SAM), the initialisation rules for those attributes (FDP_ACI), and (if any) the default mechanisms for those attributes (FDP_ACI)[32].

aa) *[FDP_ACF]* Guidance on the safe and effective use of the mechanisms[33].

ab) *[FAU_STG]* A list the conditions under which loss of audit data due to system failure shall be enumerated and the potential number of audit events lost should be documented[34].

AGD_ADM.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_ADM.1.2E: The evaluator shall confirm that the installation procedures result in a secure configuration.

AGD_USR.1: User Guidance

AGD_USR.1.1D: The developer shall provide user guidance.

AGD_USR.1.1C: The user guidance shall describe the TSF and interfaces available to the user.

AGD_USR.1.2C: The user guidance shall contain guidelines on the use of security functions provided by the TOE.

AGD_USR.1.3C: The user guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4C: The user guidance shall describe the interaction between user-visible security functions.

AGD_USR.1.5C: The user guidance shall be consistent with all other documentation delivered for evaluation.

AGD_USR.1.6C: *The user guidance shall provide guidance on:*

a) *[FIA_UAU]* Use of the TSF authentication mechanism(s).

[27] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.
[28] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.
[29] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.
[30] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.
[31] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.
[32] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.
[33] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.
[34] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.

b) *[FIA_UID]* How to identify themselves to the TOE.

c) *[FDP_ACC.1]* Guidance with respect to each access control policy satisfying a FDP_ACC component. Documentation shall be provided for end-users, authorized administrative users, or both, as appropriate for the nature of the objects and operations controlled by the policy[35].

d) *[FDP_ACF]* Information detailing what is the basis of mediation, what is the precedence of mediation when more than one conclusion could be reached given a set of attributes, etc.[36].

e) *[FDP_SAM.2]* How security attributes associated with objects, users, or subjects are modified.[37]

AGD_USR.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

[35] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.
[36] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.
[37] A CCOR concerning this element is currently being reviewed. [PPTeam54]

5.2.5 Life Cycle Support Components

ALC_DVS.1: Identification of Security Measures

ALC_DVS.1.1D: The developer shall produce development security documentation.

ALC_DVS.1.1C: The development security documentation shall describe the physical, procedural, personnel, and other security measures that are used to protect the confidentiality and integrity of the TOE during its development.

ALC_DVS.1.2C: The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

ALC_DVS.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DVS.1.2E: The evaluator shall check whether the security measures are being applied.

5.2.6 Tests Components

ATE_COV.2: Complete Coverage - Rigorous

ATE_COV.2.1D: The developer shall provide an analysis of the test coverage.

ATE_COV.2.1C: The analysis of the test coverage shall demonstrate that the tests identified in the test documentation cover the TSF.

ATE_COV.2.2C: The analysis of the test coverage shall demonstrate the correspondence between the security functions and the tests identified in the test documentation.

ATE_COV.2.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_DPT.2: Testing - High Level Design

ATE_DPT.2.1D: The developer shall provide the analysis of the depth of testing.

ATE_DPT.2.1C: The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TOE operates in accordance with the functional specification, and high level design of the TSF.

ATE_DPT.2.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_FUN.1: Functional Testing

ATE_FUN.1.1D: The developer shall test the TSF and document the results.

ATE_FUN.1.2D: The developer shall provide test documentation.

ATE_FUN.1.1C: The test documentation shall consist of test plans, test procedure descriptions, and test results.

ATE_FUN.1.2C: The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE_FUN.1.3C: The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function.

ATE_FUN.1.4C: The test results in the test documentation shall show the expected results of each test.

ATE_FUN.1.5C: The test results from the developer execution of the tests shall demonstrate that each security function operates as specified.

ATE_FUN.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2: Independent Testing - Sample

ATE_IND.2.1D: The developer shall provide the TOE for testing.

ATE_IND.2.1C: The TOE shall be suitable for testing.

ATE_IND.2.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2E: The evaluator shall test the TSF to confirm that the TSF operates as specified.

ATE_IND.2.3E: The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

5.2.7 Vulnerability Assessment Components

AVA_MSU.1: Misuse Analysis - Obvious Flaws

AVA_MSU.1.1D: The developer shall document an analysis of the guidance documentation for conflicting and incomplete guidance.

AVA_MSU.1.2D: The developer shall ensure that the guidance documentation contains no misleading or unreasonable guidance.

AVA_MSU.1.1C: The analysis documentation shall provide a rationale that demonstrates that the guidance is not conflicting and is complete.

AVA_MSU.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_MSU.1.2E: The evaluator shall determine that there is no misleading or unreasonable guidance in the guidance documentation.

AVA_MSU.1.3E: The evaluator shall repeat any procedures in the guidance documentation to ensure that they produce the documented results.

AVA_SOF.1: Strength of TOE Security Function Evaluation

AVA_SOF.1.1D: The developer shall identify all TOE security mechanisms for which a strength of TOE security function analysis is appropriate.

AVA_SOF.1.2D: The developer shall perform a strength of TOE security function analysis for each identified mechanism.

AVA_SOF.1.1C: The strength of TOE security function analysis shall determine the impact of the identified TOE security mechanisms on the ability of the TOE security functions to counter the threats.

AVA_SOF.1.2C: The strength of TOE security function analysis shall demonstrate that the identified strength of the security functions is consistent with the security objectives of the TOE.

AVA_SOF.1.3C: Each strength claim shall be either basic, medium, or high.

AVA_SOF.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_SOF.1.2E: The evaluator shall confirm that all TOE security mechanisms requiring a strength analysis have been identified.

AVA_SOF.1.3E: The evaluator shall confirm that the strength claims are correct.

AVA_VLA.1: Developer Vulnerability Analysis

AVA_VLA.1.1D: The developer shall perform and document an analysis of the TOE deliverables searching for obvious ways in which a user can violate the TSP.

AVA_VLA.1.2D: The developer shall document the disposition of identified vulnerabilities.

AVA_VLA.1.1C: The evidence shall show, for each vulnerability, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA_VLA.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VLA.1.2E: The evaluator shall conduct penetration testing, based on, but not limited to[38], the developer vulnerability analysis, to ensure obvious vulnerabilities have been addressed.

[38] A CCOR has been submitted for this modification to the CC [c2pp97.2].

Chapter 6

CAP PP Rationale and Application Notes

This chapter provides the rationale for the selection, creation, and use of the security policies, objectives, and components. Section 6.1 provides the rationale for the existence of the security objectives based upon the stated security policies while Section 6.2 provides the lower-level rationale for the existence of functional and assurance components based upon the stated security objectives. Section 6.2 provides an analysis that maps given security objectives to components as well as mapping given components to security objectives. In providing a mapping in both directions for the components and objectives, there is gained assurance that the objectives were entirely met. This is further detailed in Section 6.2.

In addition to providing a complete rationale, Chapter 6 also provides the necessary application notes needed to understand how a TOE must meet the stated security objectives. These application notes provide additional information about a particular family/component/element that a developer or evaluator may need in order to fully understand how the component is to be applied.

6.1 Security Objectives Rationale

This section provides a rationale for the existence of each threat, policy statement, security objective, and component that comprise the protection profile.

6.1.1 Complete Coverage - Threats

The TOE security objectives have been derived exclusively from statements of organizational security policy, and therefore, there are no explicitly defined threats countered by this profile.

6.1.2 Complete Coverage - Policy

This section provides evidence demonstrating coverage of the Organizational Security Policy by both the IT and Non-IT security objectives. Table 6.1 shows this objective to policy mapping, and the table is followed by a discussion of the coverage for each Security Policy.

Organizational Security Policy	Security Objectives
P.ACCESS_ENFORCE	O.CONTROLLED_ACCESS
P.ACCESS_RULES	O.DAC_ENFORCE
P.ACCOUNT	O.ACCOUNT
P.AUTH	O.ACCESS O.AUTH O.REUSE O.SYS_ARCH
P.AUTH_PROTECT	O.ACCESS
P.KNOWN	O.ACCESS O.CREDEN
P.OPERATIONAL_ASSURE	O.OPERATIONAL_ASSURE
P.MANAGE	O.ACCT_MANAGE O.INSTALL O.MANAGE O.PHYSICAL

Table 6.1 - Mapping of Security Objectives to Organizational Security Policy

The following discussion provides detailed evidence of coverage for each statement of organizational security policy:

P.ACCESS_ENFORCE

Discretionary security controls are required to ensure that only selected users or groups of users may obtain access to data (e.g., based on a need-to-know).

The O.CONTROLLED_ACCESS objective supports this policy by allowing users to manage the access to and sharing of information that they control.

P.ACCESS_RULES

Given identified subjects and objects, there must be a set of rules that are used by the TOE to determine whether a given subject can be permitted to gain access to a specific object.

The O.DAC_ENFORCE objective supports this policy by requiring that the TOE have the capability to enforce a discretionary access control mechanism.

P.ACCOUNT

The TOE must ensure that all TOE users can subsequently be held accountable for their security-relevant actions.

The O.ACCOUNT objective supports this policy by requiring the TOE to have an audit capability that protects recorded information associated with individual TOE users, and that provides some degree of selectivity of that recorded information.

P.AUTH

Each access to information must be mediated based on who is accessing the information and information they are authorized to access.

This policy is supported through a combination of objectives that address the existence of authorizations associated with individual TOE users, as specified in O.AUTH; the prevention of unauthorized access to residual information in the TOE, as specified by the O.REUSE objective; the definition of the set of resources for which the TOE provides access mediation, as specified in the O.SYS_ARCH objective; and, the ability for the TOE to identify the user requesting the information, as specified in the O.ACCESS objective.

P.AUTH_PROTECT

The identification and authorization information must be securely maintained by the TOE and be associated with every active element that performs some security-relevant action in the TOE.

The O.ACCESS objective supports this policy by requiring the TOE to have the capability to prevent unauthorized users from accessing the TOE and its resources.

P.KNOWN

Legitimate users of the TOE must be identified before TOE access can be granted.

Support of this policy requires the TOE to limit access to the TOE, as specified by the O.ACCESS objective. Additionally, the protection of access information with respect to the general operating environment must occur to ensure the legitimate TOE access information is not compromised. This is addressed by the O.CREDEN non-IT security objective.

P.OPERATIONAL_ASSURE

Features must be available that can be used to periodically validate the correct operation of the on-site hardware and firmware elements of the TSF.

This policy is supported by isolation and protection of the TSF from external sources, as specified by O.ISOLATE objective. Additional support for this policy is provided through assurance in the correct operation of the TOE by the non-existence of obvious flaws, as in the O.FLAW objective. Finally, the ability to provide these assurances throughout the lifecycle of the TOE supports the continuous protection aspect of the policy, as stated in the O.LIFECYCLE_ASSURE objective.

P.MANAGE

The TOE is managed by authorized users.

This policy is covered by the existence of the necessary functions and documentation to manage the TOE, as specified by the O.MANAGE and O.MANAGE_DOCS objectives respectively.

Additionally, this policy is supported by the correct delivery, installation and operation of the TOE, and by safeguards in place to protect the physical components of the TOE. The O.INSTALL and O.PHYSICAL non-IT objectives, respectively, provide support for this policy.

6.2 Security Requirements Rationale

This section provides evidence supporting the combined internal consistency and completeness of the functional components and EAL3 that comprise the CAP PP.

6.2.1 Internal Consistency of Requirements

This section describes the mutual supportiveness and internal consistency of the components selected for this profile. These properties are discussed for both functional and assurance components.

The functional components were all selected from pre-defined CC components. The use of component refinement (refer to Table 5.1) was accomplished in strict accordance with CC guidelines.

Assignment, selection, and refinement operations were carried out among components using consistent computer security terminology. This helps to avoid the ambiguity associated with interpretations of meanings of terms between related components.

Multiple instantiation of identical or hierarchically-related components was used to clearly state the required functionality that must exist in a TOE conformant with this profile.

The refinement operation was used to tailor selected assurance components of EAL3. However, the entirety of these refinements were for the single purpose of more clearly enumerating the implied documentation requirements resulting from the selection of particular functional requirements, as defined in the CC Part 2 Annexes.

6.2.2 Complete Coverage - Objectives

This section demonstrates that the functional components and EAL selected for this profile provide complete coverage of the defined security objectives. The mapping of components to security objective is depicted in Table 6.2.

Security Objective	Supporting Functional Components			
	FAU	FDP	FIA	FPT
O.ACCESS			FIA_ADA.3 FIA_ADP.1 FIA_ADP.2 FIA_UAU.8 FIA_UID.2 FIA_UID.3	

Security Objective	Supporting Functional Components			
	FAU	FDP	FIA	FPT
O.CONTROLLED_ACCESS		FDP_ACF.1 FDP_SAM.2	FIA_ATD.1	FPT_REV.1
O.DAC_ENFORCE		FDP_ACC.1 FDP_ACI.1		
O.ACCOUNT	FAU_GEN.1 FAU_GEN.2 FAU_PRO.2 FAU_SAR.1 FAU_SEL.1 FAU_STG.3		FIA_ADA.3 FIA_ATA.1 FIA_ATD.1 FIA_UAU.8 FIA_UID.2 FIA_UID.3 FIA_USB.1	
O.ACCT_MANAGE	FAU_MGT.1 FAU_MGT.2 FAU_PRO.2 FAU_SAR.1 FAU_SEL.1			
O.OPERATIONAL_ASSURE				FPT_AMT.1
O.SYS_ARCH		FDP_ACC.1		FPT_RVM.1 FPT_SEP.1
O.ISOLATE				FPT_RVM.1 FPT_SEP.1
O.REUSE		FDP_RIP.3		
O.MANAGE	FAU_MGT.1 FAU_MGT.2			FPT_TSA.1
O.AUTH		FDP_ACF.1 FDP_SAM.2	FIA_ATD.1	FPT_REV.2

Table 6.2 - Mapping of Security Objectives to Selected Components

The following discussion provides detailed evidence of coverage for each security objective:

O.ACCESS

The TOE must ensure that only authorized users gain access to the TOE and its resources.

Users authorized to access the TOE are defined using a restricted function [FIA_ADA.3]. To ensure authorized access to the TOE, all forms of authentication data are protected [FIA_ADP.1, FIA_ADP.2]. Identification and Authentication functions ensure that only authorized users access the TOE [FIA_UID.3, FIA_UAU.8].

O.CONTROLLED_ACCESS

The TOE must allow users to specify access control and sharing of objects based on identified individuals or groups of individuals.

An individual TOE user's access to objects must be based upon enforced access rules, [FDP_ACF.1]. The TOE must provide the capability to specifically grant or deny access to an object at the granularity of an individual user. [FDP_ACF.1]. Access attributes must be modifiable so that access to and sharing of objects is continuously controlled by the user [FDP_SAM.2]. Access control decisions are based upon user security attributes [FIA_ATD.1]. Users must be able to revoke access to objects they control [FPT_REV.1].

O.DAC_ENFORCE

The TOE must include provisions for the enforcement of discretionary access control rules.

Discretionary access control must have a defined scope of control [FDP_ACC.1]. Protection of named objects must be continuous, starting from object creation [FDP_ACI.1].

O.ACCOUNT

The TOE must provide the capability to selectively keep and protect audit information associated with individual users.

Security-relevant actions must be defined, auditable [FAU_GEN.1], and capable of being associated with individual TOE users [FAU_GEN.2]. Audit records must be stored and protected for later review [FAU_STG.1, FAU_PRO.2]. The TOE must provide the capability to selectively audit and selectively review security-relevant events of an individual TOE user [FAU_SEL.1, FAU_SAR.1]. Authentication data is required to uniquely identify a TOE user [FIA_ADA.3]. User attributes (e.g., group ID) can be associated with security-relevant events [FIA_ATD.1, FIA_ATA.1]. The TOE must be able to identify and authenticate individual TOE users [FIA_UID.3, FIA_UAU.8]. A user's attributes must be correctly associated with a subject acting on behalf of that user [FIA_USB.1].

O.ACCT_MANAGE

The TOE must provide the capability for an authorized user to access and evaluate accountability information by a secure means.

Management functions must be provided to support accountability functions [FAU_MGT.1]. Audit trail saturation must be brought to the attention of the authorized user [FAU_MGT.2]. Accountability data must be protected from access by unauthorized users [FAU_PRO.2]. Support must be provided for access to and evaluation of accountability data [FAU_SEL.1, FAU_SAR.1].

O.OPERATIONAL_ASSURE

Assurance must be provided that correct implementation and enforcement of the policy exists throughout the TOE's life-cycle.

This objective is met through the assurance requirements defined in EAL3. Additionally, the TOE must provide the capability to demonstrate correct operation of the TSF's underlying abstract machine [FPT_AMT.1].

O.SYS_ARCH

The TOE must provide a defined set of resources that the TSF controls.

The TOE must enforce the security policy on a defined set of resources [FDP_ACC.1]. The TSF must maintain its own execution domain [FPT_SEP.1]. Access enforcement to the defined set of resources controlled by the TSF must not be circumventable [FPT_RVM.1].

O.ISOLATE

The TSF must maintain a domain for its execution that protects itself from external interference or tampering.

The TSF and protected resources must be distinct parts of the TOE, and the TSF must be protected from non-TSF entities [FPT_SEP.1]. Access enforcement to the defined set of resources controlled by the TSF must not be circumventable [FPT_RVM.1].

O.REUSE

The TOE must ensure that residual information is made unavailable for unauthorized reuse.

Residual information associated with defined objects in the TOE must be purged prior to the reuse of the object containing the residual information [FDP_RIP.1, FDP_RIP.2].

O.MANAGE

The TOE must provide all the functions and facilities necessary to support the authorized users that are responsible for the management of TOE security.

Accountability functions and data must be managed [FAU_MGT.1, FAU_MGT.2]. Administrative functions must be provided to support the security management of the TOE [FPT_TSA.1].

O.AUTH

Trusted users have associated authorization(s) that allow access to data not intended to be accessed by untrusted users.

The TOE must include a user’s authorization attributes in access control decisions. [FDP_ACF.1]. Authorization attributes must be modifiable by authorized users [FDP_SAM.2]. The TOE must associate authorizations with each TOE user [FIA_ATD.1]. The TOE must provide the capability to revoke user authorization attributes [FPT_REV.2].

6.2.3 Functional Requirements Rationale and Application Notes

This section presents a listing of each functional component, its application notes, and its rationale. These notes provide an explanation of terminology and expected interpretations of meanings, and should aid in the analysis of requirement satisfaction for users of the profile. This material presents each component's contribution to satisfying one or more security objectives.

FAU_GEN.1: Audit Data Generation

FAU_GEN.1.1: The TSF shall be able to generate an audit record of the following auditable events:

a) Startup and shutdown of the audit functions.

b) All auditable events for the *basic* level of audit, as defined in all functional components included in the PP/ST; and

c) Based on all functional components included in the PP/ST,[1]

- *The use of security authorizations (e.g., privileges) to override enforcement of the SFP;*
- *All actions by administrators;*
- *Establishment of a path that connects a user to another users process;*
- *Successful and unsuccessful attempts to specify the granting or denying of access to an object; and*
- *[assignment: other auditable events].*

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of event, type of event, subject identity, and *success or failure* of the event.

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST,

- *For Identification & Authentication events, the origin (e.g., terminal ID) of the request;*
- *For introduction of an object into a user's address space, and for object deletion, the object identity;*
- *For administrator changes to security-relevant databases, the new values for changed items;*
- *For administrator actions in a TOE with multiple operator consoles, the identity of the console from which the auditable event originated; and*
- *[assignment: other audit relevant information].*

Application Notes

The "basic" level of auditing events was selected from the functional families of components selected for this profile. In addition to these events, the use of security authorizations (e.g., privileges) and all actions by security administrators were also specified as generic, security-relevant events. In addition, the assignment operation was carried forward as a reminder to ST authors to supplement the profile with system-specific, security-relevant events. The ST should also demonstrate explicit coverage of all event types by mapping to implemented event types. Note that manual logging of events is sometimes acceptable to meet audit requirements, but acceptability of this practice has been

[1] These are additional auditable events not covered in a) or b) above.

low and it is generally discouraged. The following list provides extra detail and interpretations of the types of events that are intended to be generated.

- FAU_MGT.1. Operations on the audit trail include modification, deletion, etc. As such, it may overlap somewhat with the events under FAU_PRO; however; the intent here is more focused on administrative operations.
- FAU_MGT.2. In the functional component, “saturation” is specified as a pre-defined limit that may be configurable. This event would record the occurrence of saturation as defined under the specific ST.
- FAU_PRO. Actions that attempt to access the audit trail should be recorded. It is possible that this is accomplished by analysis of more generic audit records (e.g., file write attempts).
- FAU_SEL. Modifications include changing parameters for pre-selection auditing functions (e.g., an audit “mask”). The system administrator shall be able to audit based on user identity.
- FDP_ACF.1. Operations on objects include the following: use of access rights to bypass policy checks, introduction of an object into the user’s address space, deletion of an object, and production of printed output. Some operations may not be security-relevant (e.g., file requests that result in “object not found”) and so do not require auditing. These types of events should be specified in the ST. In combination with recording of event “failures” under FAU_GEN.1.2, auditing this type of event allows discovery of repeated attempts to bypass protection mechanisms.
- FDP_ACF.1. Operations that grant or deny access are system-dependent and so must be identified in the ST. For example, the passage of an unnamed pipe descriptor through a socket to a different subject would be considered an access grant, and must be auditable.
- FDP_ACI. Auditing of these events is complementary to those of FDP_SAM.
- FDP_SAM. Modifications include changes to a subject’s and/or object’s security attributes. The auditing of the distribution and revocation of access rights and capabilities would be considered this type of event.
- FIA_ADA. e.g. initializing user accounts and changing user passwords.
- FIA_ADP. Complementary to FIA_ADA events, and possibly achievable by indirect means such as file “opens.”
- FIA_UAU. Potentially coincidental with FIA_UID events.
- FIA_UID. Potentially coincidental with FIA_UAU events. **User ID's given for failed login attempts need not be audited.**[2]
- FIA_USB. These events maintain the mapping between user and subject identities (e.g., when a process is created).
- FPT_AMT. Audit of automatic abstract machine testing events and test failures may be satisfied implicitly by audit records of other events (e.g., start-up) which imply their occurrence, and in general, need to be audited only where feasible.[3]
- FPT_TSA. Security-relevant administrative functions include all actions taken by the operator, system administrator, and system security administrator. This component also addresses security-relevant events that result from the use of TOE interfaces not advertised for general use.

[2] A CCOR has been submitted for this issue [c2pp97.3].

[3] A CCOR has been submitted for this issue [c2pp97.4].

For the audit record data requirements specified under FAU_GEN.1.2, data elements that are traditionally recognized as integral to the audit function (e.g., date and time of the event) are specified. The selection of "success or failure" to be recorded should be interpreted in a security-relevance mode: if the success, failure, or either success or failure of an event is security-relevant, then that occurrence must be captured in an audit record. These interpretations must be made explicit in the ST. Additional information for specific events has been specified under the assignment statement of this element. This data has been determined to be security-relevant for these events through experience in the practice of audit. In addition, the assignment operation was carried forward as a reminder to ST authors to supplement the profile with system-specific, security-relevant audit record data.

Rationale

This component supports O.ACCOUNT by specifying the detailed, security-relevant events and data that the audit mechanism must be capable of generating and recording. These security-relevant events are defined on a per-family and per-component bases in the CC. Thus, much of the analysis for security relevance is "built-in" and is incorporated mechanically when constructing the profile. However, the component has been made extensible and both the auditable event list and required audit data lists have been extended to incorporate traditional auditing knowledge and practice.

The "basic" level of auditing was selected as best representing the "mainstream" of contemporary audit practice. Significantly more (and more precise) information is recorded at this level than at the "minimum" level, yet it avoids the overhead and implementation burdens of the more extreme "detailed" level. Thus, this selection is somewhat of a compromise, aimed at achieving a robust audit mechanism at a reasonable cost.

FAU_GEN.2: User Identity Generation

FAU_GEN.2.1: The TSF shall be able to associate any auditable event with the identity of the user that caused the event.

Application Notes

Auditable events are specified under FAU_GEN.1, which is a dependency for this component. It should be noted that failed login attempts may not associate the event with the identity of a user. The requirement for user identities for this profile is specified under FIA_UID.3, which is an indirect dependency for this component.

Rationale

O.ACCOUNT calls for individual accountability (i.e., “TOE users”) whenever security-relevant actions occur. This component requires every auditable event to be associated with an individual user. Since FAU_GEN.1 specifies that all security-relevant events (e.g., those invoking the discretionary security policy) must be auditable, individual accountability can be established for these events.

FAU_MGT.1: Audit Trail Management

FAU_MGT.1.1: The TSF shall provide the authorized administrator with the ability to *create, delete, and empty* the audit trail.

Application Notes

The selection of “create, delete, and empty” functions for audit trail management reflect common management functions. These functions should be considered generic—any other audit administration functions that are critical to the management of a particular audit mechanism implementation should be specified in the ST.

Rationale

FAU_MGT.1 specifies basic, audit trail management functions. Maintenance requires creating, deleting and emptying the audit trail. This component supports the O.MANAGE and O.ACCT_MANAGE objectives. In order for audit trail review to occur, audit data must first be collected. This collection activity requires the support functionality provided by this component. Limiting use of these functions to authorized administrators explicitly supports the O.ACCT_MANAGE objective.

FAU_MGT.2: Audit Trail Saturation Control

FAU_MGT.2.1: The TSF shall generate an alarm to the authorized administrator if the size of the audit data in the audit trail exceeds a *[assignment: pre-defined limit]*.

Application Notes

For this component, an “alarm” is to be interpreted as any clear indication to the administrator that the pre-defined limit has been exceeded. The ST author must state the pre-defined limit that triggers generation of the alarm. The limit can be stated as an absolute value (e.g, full audit trail), or as a value that represents a percentage of audit trail capacity (e.g., audit trail 75% full). This is important to avoid disruption of service, in light of the requirements under FAU_STG.3 that prevent user activity when storage capacity is reached.

Rationale

This component supports the goals of O.MANAGE and O.ACCT_MANAGE. It helps ensure that situations involving potential loss of audit data and/or service disruption are brought to an administrator’s attention. Audit trail saturation is a common failure mode for audit mechanisms and this function greatly enhances the ease-of-use concerns. Acceptable pre-defined limits are somewhat subjective and depend upon the requirements of the operational environment. Therefore, it is appropriate to defer specification of this limit to the ST.

FAU_PRO.2: Extended Audit Trail Access

FAU_PRO.2.1: The TSF shall restrict full access to the audit trail to the authorized administrator.

FAU_PRO.2.2: The TSF shall provide only authorized users with the capability to read *[assignment: list of audit information]* from the audit trail.

Application Notes

Authorized administrators are those responsible for managing the audit trail. They are required to have full access in order to perform normal maintenance functions such as creating, deleting, and emptying the audit trail. Administrative access is assumed to be non-malicious. Authorized users are those that are explicitly or implicitly authorized to read all or part of the audit trail. What constitutes an "authorized user" (e.g., all users) must be described in the ST. Additional constraints, such as restricting authorized users to reading only their own audit records, must be elaborated in the ST.

Rationale

O.MANAGE calls for the capability for administrators to access the audit trail securely. This component directly supports this objective by limiting complete access to administrators. Access by authorized users is limited to "read" access. This limitation prevents anyone other than administrators from altering audit trail information, thus preserving accountability, as stated in the O.ACCOUNT objective. Allowing authorized users the capability of potentially reading part of the audit trail may be appropriate and secure for some environments and implementations.

FAU_SAR.2: Extended Audit Review

FAU_SAR.2.1: The TSF shall provide audit review tools, with the ability to view the audit data.

FAU_SAR.2.2: The TSF shall restrict full use of the audit review tools to the authorized administrator.

Application Notes

Audit review tools are those that read the system’s audit trail and transforms the event records into a human-readable representation. In some instances, it may be acceptable to allow an authorized user to a view a subset of the audit data; e.g., audit records generated by that user. The developer is responsible for describing to evaluators what constitutes an “authorized user” (e.g., all users).

Rationale

This component supports both the O.ACCOUNT and O.ACCT_MANAGE objectives. Accountability is supported by providing a means for administrators to review audit trail contents associated with an individual TOE user.

FAU_SAR.3: Selectable Audit Review

FAU_SAR.3.1:[4] The TSF shall provide audit review tools with the ability to perform *[selection: searches, sorting]*[5] of audit data based on *user identity and [assignment: other logical operations] if audit collection data cannot be restricted on the basis of user identity*.[6]

Application Notes

The intent is to provide for selection of audit information on the basis of user identity. Either pre- or post-selection is acceptable.

Rationale

This component supports both the O.ACCOUNT and O.ACCT_MANAGE objectives. Accountability is supported by providing a means for administrators to selectively review audit trail contents associated with an individual TOE user. Audit data can be searched for only those records applying to a particular individual.

[4] A CCOR concerning a slight wording change is currently being processed. [PPTeam33]

[5] A CCOR concerning this change has been submitted to the CC [c2pp97.5].

[6] A CCOR concerning this change has been submitted to the CC [c2pp97.10].

FAU_SEL.1: Selective Audit

FAU_SEL.1.1: The TSF shall be able to include or exclude auditable events from the set of audited events based on *one or more of* the following attributes:

a) *User identity; and*

b) *[assignment: list of additional attributes]* that audit selectivity is based upon.

if the audit review tools cannot search or sort data on the basis of user identity[7].

Application Notes

The intent is to provide for selection of audit information on the basis of user identity. Either pre- or post-selection is acceptable.

Auditable events are specified in FAU_GEN.1. The term “user identity” specifies selectivity based upon named individuals. The ST author must state the additional attributes that audit selectivity may be based upon (e.g., object identity, type of event), if any.

Rationale

This component provides the critical functionality of allowing administrators to control what events are to be actually audited in an operational environment. It allows audit events to be selected on the basis of individual users, as called for in O.ACCOUNT. Besides allowing an individual-level granularity for auditing, this component provides necessary functionality for limiting the audit trail size and rate of growth, supporting O.ACCT_MANAGE.

[7] A CCOR concerning this change has been submitted to the CC [c2pp97.10].

FAU_STG.3: Prevention of Audit Data Loss

FAU_STG.3.1: The TSF shall store generated audit records in a permanent audit trail.

FAU_STG.3.2: The TSF shall limit the number of audit records lost due to system *audit storage exhaustion and failure*.

FAU_STG.3.3: In the event of audit storage exhaustion, the TSF shall be capable of *preventing* the occurrence of auditable actions, except those taken by the authorized administrator.

Application Notes

The audit trail is considered to be the data repository of all audit event records generated by the system. The term “permanent” implies persistent storage and also implies the duration of validity of the audit information. Limiting the loss of audit records due to “audit storage exhaustion and failure” requires a specific limit (maximum) be delineated for the common failure modes. The actual limit of potential audit data losses should be described in the ST and should be minimized to a reasonable degree under the implementation. The selection of “preventing” auditable actions if audit storage is exhausted is minimal functionality; providing a range of configurable choices (e.g., ignoring auditable actions and/or changing to a degraded mode) is allowable and preferred, but not required.

Rationale

This component supports O.ACCOUNT in three ways. First, the existence of a permanent audit trail supports storage of user audit information, the source of individual accountability. Second, limiting the loss of audit data in exceptional conditions minimizes the loss of this audit data to its practical limits. Third, the prevention of user activity in the case of storage exhaustion protects accountability by ensuring the disruption of auditable actions. These three properties serve to enhance the completeness of accountability data.

FDP_ACC.1: Subset Object Access Control

FDP_ACC.1.1: The TSF shall enforce the *Discretionary Access Control Policy* on *named individuals, subjects acting on behalf of named individuals, groups of named individuals, and [assignment: list of named objects]* for *[assignment: operations among subjects and named objects covered by the Discretionary Access Control Policy]*.

Application Notes

The ST author must explicitly list the named objects that exist in the TOE. Additionally, during the course of the evaluation against the TOE, the evaluation team may determine that the list be modified. Upon completion of the TOE evaluation, the list of named objects must be complete and consistent with the named objects evaluated by the evaluation team.

The operations among subjects and named objects must explicitly define all relationships between subjects and named objects in the TOE, and must be consistent with the list of named objects defined in the earlier assignment.

Not all named objects protected by DAC are required to use the same rules or support the same flexibility of control. When appropriate, the ST author may use multiple instantiations of FDP_ACC.1 and related DAC-policy components in the product ST to reflect different types of objects and operations. In some cases, a careful specification of the rules in FDP_ACF.1 will suffice.

A set-ID mechanism may be part of an acceptable DAC implementation.

Rationale

This component is mapped to the O.DAC_ENFORCE and O.SYS_ARCH objectives. It defines the scope of access control policy enforcement, which for a CAP PP-conformant TOE, does not require all objects to be subject to protection requirements.

FDP_ACF.1: Single Security Attribute Access Control

FDP_ACF.1.1: The TSF shall enforce the *Discretionary Access Control Policy* to objects based on *the following subject attributes:*

a) *named individuals;*

b) *defined groups of individuals;*

c) *[assignment: list of subject authorizations].*

FDP_ACF.1.1: The TSF shall enforce the *Discretionary Access Control Policy* to objects based on *the following named object attributes:*

a) *access rights*.

FDP_ACF.1.2: The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

a) *[assignment: rules governing access among subjects and named objects using operations on named objects].*

Application Notes

The terminology of "named individual" includes subjects acting on behalf of a named individual. The ST author must explicitly state the rules for each operation between subjects and named objects in the TOE.

It is not required that all users have the capability to control the sharing of objects. It is sufficient that only the system administrators assign access to objects. The Administrators Guidance shall clearly identify the roles or user types (e.g., system administrator) who can control sharing.

An object for which the TSF unconditionally permits all subjects "read" access shall be considered a public object, provided that only the TSF or privileged subjects may create, delete, or modify the object. No access checks or auditing are required for "read" accesses to such objects. Attempts to create, delete, or modify such objects require access checks and are auditable.

Discretionary Access Control enforcement mechanisms that directly depend upon passwords shall not be considered sufficient.

An authenticated user that has been authorized (i.e., granted access) to connect to another user's process shall be allowed to do so provided that the establishment of the connection path is auditable.

Single-user granularity can be obtained without explicit single-user inclusion or exclusion. For example, the use of the UNIX permission bits mechanism can be used to achieve single-user granularity.

Rationale

This component is mapped to the O.CONTROLLED_ACCESS and O.AUTH objectives. This component allows specification of the subject and named object attributes used as a basis for the rules enforced by the TOE for user-specified access control and sharing of objects.

FDP_ACI.1: Static Attribute Initialization

FDP_ACI.1.1: The TSF shall enforce the *Discretionary Access Control Policy* to provide *restrictive* default values for the object security attributes that are used to enforce the *Discretionary Access Control Policy*.

FDP_ACI.1.2: The TSF shall allow the specification of alternate initial values to override the default values when the object is created.

Application Notes

A CAP PP-conformant TOE must provide protection by default for all objects at creation time. This may be done through the enforcing of a restrictive default access control on newly created objects or by requiring the user to explicitly specify the desired access controls on the object at its creation. In either case, there shall be no window of vulnerability through which unauthorized access may be gained to newly created objects.

Rationale

This component maps to the O.DAC_ENFORCE objective. The requirement for restrictive default values enforces protection of objects at the instant of object creation.

FDP_RIP.3: Full Residual Information Protection on Allocation

FDP_RIP.3.1: The TSF shall ensure that upon the allocation of a resource to all objects any previous information content is made unavailable.

Application Notes

These requirements apply to all sharable objects and their attributes (i.e., objects to which DAC are applied), as well as other system resources (e.g., stacks, process memory), and includes encrypted representations of information.

Clearing the information content of resources on deallocation from objects is sufficient to satisfy this requirement, since unallocated resources will not accumulate new information until they are allocated again.

Rationale

These components are mapped to the O.REUSE objective.

FDP_SAM.2: User Attribute Modification

FDP_SAM.2.1: The TSF shall enforce *the Discretionary Access Control Policy* to provide authorized users with the ability to modify:

a) *named object access control specifications, based on named individual and/or defined groups of named individuals.*

b) *[assignment: other security attributes]*

Application Notes

The ST author must state the components of the access control specification that may be modified, and must state any restrictions that may exist for a type of authorized user and the components of the access control specification that the user is allowed to modify.

The ST author must state any additional modifiable security attributes that exist in the TOE, or provide a statement that no additional modifiable attributes exist.

Rationale

This component maps to the O.CONTROLLED_ACCESS and O.AUTH objectives. This component defines the scope of attributes that a user can modify when controlling the access to and sharing of objects.

FIA_ADA.3: Expanded User Authentication Data Administration

FIA_ADA.3.1: The TSF shall provide functions for initializing and modifying user authentication data related to *[assignment: identified authentication mechanism]*.

FIA_ADA.3.2: The TSF shall restrict the use of these functions on the user authentication data for any user to the authorized administrator.

FIA_ADA.3.3: The TSF shall allow authorized users to use these functions to modify their own authentication data in accordance with the TSP.

Application Notes

User authentication data refers to information that users must provide to authenticate themselves to the TSF. Examples include passwords, personal identification numbers, and fingerprints. User authentication data does not include the users identity.

An authorized user refers to an authenticated user of the system.

The ST author must specify the authentication mechanism that makes use of the user authentication data to verify a user’s identity.

It is not necessary that requests to modify authentication data require reauthentication of the requester’s identity at the time of the request.

This component does not exclude centralized administration of passwords or other authentication data.

Rationale

This component maps to the O.ACCESS and O.ACCOUNT objectives. An aspect of accountability is the capability to protect authentication information associated with each user identity. This component allows the system administrator full control over any user authentication data and allows an individual user control over his own authentication data in accordance with the TSP.

FIA_ADP.1: Basic User Authentication Data Protection

FIA_ADP.1.1: The TSF shall protect from unauthorized observation, modification, and destruction authentication data that is stored in the TOE.

Application Notes

The card input of batch jobs may contain human-readable user passwords. The Administrator and User Guidance documentation for the product shall explain the risks in placing passwords on card input and shall suggest procedures to mitigate that risk.

Rationale

This component maps to the O.ACCESS objective. Individual accountability cannot be maintained if the individual's authentication data is compromised.

FIA_ADP.2: Extended User Authentication Data Protection

FIA_ADP.2.1: The TSF shall protect from unauthorized observation, modification, and destruction the raw form of authentication data at all times while it resides in the TOE.

Application Notes

Unauthorized observation implies the TSF does not produce a visible display of any authentication data entered through the keyboard (e.g., echo the password on the terminal).

Rationale

This component maps to the O.ACCESS objective. Individual accountability cannot be maintained if the individual's authentication data, in any form, is compromised.

FIA_ATA.3 Extended User Attribute Administration

FIA_ATA.3.1 The TSF shall provide the ability to *display and modify* user attributes.

FIA_ATA.3.2 The TSF shall limit the ability to modify any user's attributes to only the authorised administrator.

FIA_ATA.3.3 The TSF shall allow users to modify their own attributes in accordance with the TSP.

Application Notes

This component provides administrators with full control of user's attributes, while allowing users to modify and display their attributes (e.g., primary group association, umask) as long as the TOE's security policy allows it.

Rationale

It is clear that the TOE must exhibit the ability for administrators to display and modify user's attributes in the course of maintaining and operating a secure system. This component provides the users the ability to maintain a subset of their attributes as they see fit within the constraints of the TOE's security policy. This flexibility provides users the control to manipulate their attributes and take greater interest in maintaining a secure environment.

FIA_ATD.1: User Attribute Definition[8]

FIA_ATD.1: The TSF shall provide, for each user, a set of security attributes necessary to enforce the TSP.

Application Notes

The notion of user security attributes does not include user authentication data and therefore allows user security attributes to be associated with groups of users.

Rationale

This component maps to the O.ACCOUNT, O.CONTROLLED_ACCESS, and O.AUTH objectives. The enforcement of access controls based on unique user identities and accountability of individual users requires a unique set of user attributes to explicitly identify each user. User authorizations are used by the TOE when making access control decisions.

[8] An FRG Note was written about whether to select FIA_ATD.1 or FIA_ATD.2. Since the FRG recommends using FIA_ATD.1, there is no issue here. [FRG Note 7]

FIA_UAU.8: Timing of Authentication[9]

FIA_UAU.8.1: The TSF shall allow users to perform *no actions requiring TSF mediation* before the user's claimed identity is authenticated.

FIA_UAU.8.2: The TSF shall perform the authentication of any user's claimed identity prior to performing any TSF-mediated actions on behalf of the user.

Application Notes

Single-use authentication mechanisms, such as one-time password devices, can be part of an acceptable Identification and Authentication (I&A) mechanism.

This profile treats I&A as a single function enforced at the time of authentication. Refer to FIA_UID.3.

In this context, TSF-mediation is where an access decision is made using a subject's identity (derived from an authenticated user) to make a determination if access to a protected object is allowed. TSF actions that would be acceptable before user authentication include displaying a login banner, responding to help requests with help information, and providing a mechanism to request assistance from a trusted user. Actions that would not be acceptable include access to any protected file system object and sending or receiving messages from or to other users or terminals.

Rationale

This component maps to the O.ACCESS and O.ACCOUNT objectives. The CC philosophy behind this requirement is to allow the PP/ST author to differentiate between TSF-mediated events that a user could perform without authenticating their identity to the TSF (via assignment in FIA_UAU.8.1), and all other TSF-mediated events that require authentication of the user's identity.

A CAP PP-conformant TOE requires that prior to having been identified and authenticated by the TSF, a user communicating with the TSF may be allowed to perform only those actions that would not require TSF mediation. Therefore, the assignment in FIA_UAU.8.1 prohibits any TSF-mediated action prior to authentication. Furthermore, this profile treats I&A as a single function enforced at the time of authentication, requiring consistency in the assignments made in this component and in FIA_UID.3 (refer to FIA_UID.3). The combination of this component with FIA_UID.3 address the O.ACCESS and O.ACCOUNT objectives.

[9] An FRG Note was written about the assignment statements for this component. However, despite the FRG recommendation to change the assignment statement, nothing was changed from the original CC. [FRG Note 8]. A CCOR was written [PPTeam29] stating that there was an incorrect dependency. The team received a favorable FRG response.

FIA_UID.2: User Identification

FIA_UID.2.1: The TSF shall uniquely identify each user.[10]

Application Notes

The TSF must be able to identify the user which is associated with each action for purposes of access control and auditing. On many systems there are two representation of user identity, one which represents the name of the user and the other a numeric representation which is used internally by the TOE. In these cases the identity must be unique within each name space and a unique mapping between them exist.

Rationale

This component maps to the O.ACCOUNT and O.ACCESS objectives.

[10] A CCOR has been submitted for this change [c2pp97.7].

FIA_UID.3: Timing of Identification

FIA_UID.3.1: The TSF shall allow users to perform *only actions that do not require access to the TOE information, services, or resources that are restricted by policy on the basis of user identity or other security attributes* before identifying the user.

FIA_UID.3.2: The TSF shall identify each user before performing any other actions on behalf of the user.

Application Notes

The CC separates I&A into 2 distinct but related functions. This profile makes no such distinction, and requires I&A to be mutually enforced at the time of authentication (Refer to FIA_UAU.8).

Rationale

This component maps to the O.ACCOUNT and O.ACCESS objectives. Since I&A is treated by this profile as a single function enforced at the time of authentication, there must be consistency between the assignment made here and the assignment made in FIA_UAU.8. The combination of this component with FIA_UAU.8 addresses the O.ACCOUNT and O.ACCESS objectives.

FIA-USB.1: User-Subject Binding

FIA_USB.1.1: The TSF shall associate the appropriate user security attributes with subjects acting on behalf of that user.

Application Notes

No application note is provided for this requirement.

Rationale

This component is mapped to the O.ACCOUNT objective. The ability to account for the actions of an individual requires that the TSF be capable of properly associating the attributes of any subject acting on behalf of an individual user with the attributes of that user.

FPT_AMT.1: Abstract Machine Testing

FPT.AMT.1.1: The TSF shall provide the authorized administrator with the capability to demonstrate the correct operation of the security-relevant functions provided by the TSF's underlying abstract machine.

Application Notes

This component is necessary to meet an aspect of the O.OPERATIONAL_ASSURE objective. One aspect of life cycle assurance focuses on features and system architecture used to ensure that the security policy is uncircumventably enforced during system operation. That is, the security policy must be integrated into the hardware and software protection features of the system. An example of a step taken to provide this kind of confidence is methods for testing the operational hardware and software for correct operation.

Rationale

This component requires the TOE to have the capability to test the underlying abstract machine which the TSF relies upon. This abstract machine could be a hardware/firmware platform, or it could be some known and assessed hardware/software combination acting as a virtual machine. Examples could be testing hardware page protection, sending sample packets across a network to ensure receipt, verifying the behavior of the virtual machine interface, etc. The developer must provide a description of what hardware/software/firmware aspects of the machine the developer's test suite addresses. It is intended that the developer's test suite tests the security-relevant functionality of the TOE's underlying platform. If the developer's test suite does not test the correct operation of all functionality (e.g., the test suite covers the entire chipset and firmware) than the developer must ensure they adequately test the security-relevant portions.

The test suite used to meet this requirement can be executed in a maintenance state, as part of system start-up, on-line, or continuously. However, controls should be in place to limit access to authorized users. If the features provided by the developer to meet this requirement cannot be exercised by the purchaser of the product, the developer shall make available appropriate services to use the features as needed to meet this requirement. These services shall be available on an on-demand basis.

FPT_REV.1: Basic Revocation

FPT_REV.1.1: The TSF shall provide a capability for revocation of security attributes associated with the *named users, named objects and [assignment: list of additional resources]* within the TSC[11].

FPT_REV.1.2: The TSF shall enforce revocation *of:*

a) *Discretionary Access Control security attributes. Changes shall have an effect for new access requests.*

b) *[assignment: specification of other revocation rules].*

Application Notes

DAC policies may vary and can include immediate revocation (e.g., Multics immediately revokes access to segments) or delayed revocation (e.g., most UNIX systems do not revoke access to already opened files). DAC permission is considered to have been revoked when all subsequent access control decisions by the TSF use the new access control information. It is not required that every operation on an object make an explicit access control decision as long as a previous access control decision was made to permit that operation. It is sufficient that the developer clearly documents in end-user documentation how revocation is enforced.

Rationale

This component is required to meet an aspect of the O.CONTROLLED_ACCESS security objective. In order to meet the O.CONTROLLED_ACCESS objective, a TOE must provide a DAC mechanism that may vary and can include immediate revocation or delayed revocation.

[11] TOE Scope of Control

FPT_REV.2: Immediate Revocation

FPT_REV.2.1: The TSF shall provide a capability for revocation of security attributes associated with the *named users and [assignment: list of additional resources]* within the TSC.

FPT_REV.2.2: The TSF shall immediately enforce revocation of *security-relevant authorizations.*

Application Notes

The term “security-relevant authorization” is used here to mean a capability, assigned administratively to some user(s), to perform security-relevant operations not permitted ordinary, untrusted users (e.g., privileges, administrative roles). These authorizations should be specified in the ST. The term “authorization” is not intended to cover controls on routine access to data by untrusted users; that is handled through the DAC mechanism. Examples of authorizations include: ability to act as system administrator, ability to set audit control parameters, ability to act as system operator, ability to manipulate others’ requests in printer queues, and ability to change other users’ passwords. Many authorizations could have serious consequences if misused, so an immediate revocation method must exist, although it need not be the usual method (e.g., The usual method may be editing the trusted users profile, but the change doesn’t take effect until the user logs off and logs back on. The method for immediate revocation might be to edit the trusted users profile and “force” the trusted user to log off.). The immediate method must be specified in the ST and in administrator guidance. In a distributed environment the developer must provide a description of how the “immediate” aspect of this requirement is met.

An example of an additional resource in the assignment statement of FPT_REV.2.1 would be a system daemon or some job not necessarily associated with a named user.

Rationale

This component is required to meet an aspect of the O.AUTH security objective. In order to meet the O.AUTH objective, a TOE must have provide a mechanism for associating authorizations with trusted users, as well as a mechanism for removing the authorizations from trusted users. The mechanism for removing the authorizations from trusted users must be immediate since authorizations could have a serious security impact.

FPT_RVM.1: Non-Bypassability of TSP

FPT_RVM.1.1: The TSF shall ensure that the TSP enforcement functions are invoked and succeed before any security-related operation is allowed to proceed.

Application Notes

This element does not imply that there must be a reference monitor. Rather, this element requires that the TSF validates against the Security Function Policy (SFP) all actions between untrusted subjects and trusted objects that require policy enforcement.

The term “untrusted subject” refers to subjects untrusted with respect to any or all of the specific SFPs being enforced. A subject may be trusted with respect to one SFP and untrusted with respect to another SFP.

Rationale

FPT_RVM.1 combined with FPT_SEP.1 are mapped to the O.SYS_ARCH and O.ISOLATE security objectives. Specifically, the application notes in FPT_SEP.1.2 address the validation requirements mentioned in the application notes for this component.

FPT_SEP.1: TSF Domain Separation

FPT_SEP.1.1: The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2: The TSF shall enforce separation between the security domains of subjects in the TSC.

Application Notes

The FPT_SEP.1.1 element does not imply the presence of a reference monitor, as there is no mandatory distinct reference monitor domain. The resources of the TSF's security domain ("protected domain") and those of subjects and unconstrained entities external to the domain must be separated such that the entities external to the protected domain cannot observe or modify data structures or code internal to the protected domain. Resources controlled by the TSF may be a defined subset of the subjects and objects in the TOE.

In regard to the FPT_SEP.1.2 element, transfers between domains are controlled such that the arbitrary entry to, or return from, the protected domains is not possible. The user or application parameters passed to the protected domain by addresses are validated with respect to the protected domain's address space, and those passed by value are validated with respect to the values expected by the protected domain. The FPT_SEP.1.1 application notes should also be considered in understanding this element.

Rationale

The FPT_SEP.1.1 element is mapped to the O.SYS_ARCH and O.ISOLATE security objectives. Not only must the TOE provide a TSF, but it must also ensure that the TSF supports the isolation for objects (also covered in FPT_SEP.1.2). This element provides the basis for a reference validation mechanism-like structure. FPT_SEP.1.2 is mapped to the O.ISOLATE security objective.

FPT_TSA.1: Basic Security Administration

FPT_TSA.1.1: The TSF shall distinguish security-relevant administrative functions from other functions.

FPT_TSA.1.2: The TSF's set of security-relevant administrative functions shall include all functions necessary to install, configure, and manage the TSF; minimally, this set shall include: *[assignment: list of administrative services to be minimally supplied].*

FPT_TSA.1.3: The TSF shall restrict the ability to perform security-relevant administrative functions to specifically authorized users.

FPT_TSA.1.4: The TSF shall be capable of distinguishing the set of users authorized for administrative functions from the set of all users of the TOE.

Application Notes

All administrative actions are security-relevant. Security-relevant administrative functions include all actions taken by the operator, system administrator, and system security administrator. This component also addresses security-relevant events that result from the use of TCB interfaces not advertised for general use. This component requires that the TSF distinguish between administrative and non-administrative functions. The list of administrative services to be minimally supplied needs to be identified in the STand should include:

- creation, deletion of user accounts
- assignment or modification of any security relevant configuration data or attributes
- any administrative action requiring privilege or authorization.

Rationale

This component supports the O.MANAGE objective. In addition, accountability requires that security-relevant events be recorded. In order for that to occur, security-relevance must first be established.

6.2.4 Assurance Requirements Rationale and Application Notes

6.2.4.1 Rationale for EAL3

The CC assurance requirements generally offer confidence that the security functions of a TOE work as designed. A chosen assurance level (an EAL) is a direct statement by the PP writer (or sponsoring organization) of the level of confidence that must be present in PP compliant TOEs. This “statement” must consider the targeted environment and the selection of the functional requirements that have been chosen to be incorporated into the PP. Because the notion of confidence is abstract and difficult to objectively justify, choosing an EAL involves embracing a particular philosophy more than relying on an algorithmic approach (such as a consistency mapping).

Generally CC EAL3 was chosen as the assurance package for this profile considering the following:

a) The access control policy is discretionary and is vulnerable to various trojan horse types of attacks. Given this residual generic vulnerability, higher assurance does not seem warranted. This seems acceptable seeing as the targeted environment is benign and cooperative (see A.COOP).

b) It is envisioned that TOEs compliant with this PP will be used to process sensitive information. Given this, appropriate design analysis and testing is required to show that it is designed and operates in a manner that can protect information from accidental compromise and unsophisticated attacks.

The following discussion describes each class of the assurance requirements included in the EAL3 assurance package. All of these assurance activities are seen as vital to gain appropriate confidence that compliant TOEs are suitable to process sensitive information.

Configuration Management

The inclusion of this class of assurance requirement acknowledges that products evolve over time. The rationale for the configuration management requirements is two fold:

a) to maintain control of modifications made to the TOE;

b) to facilitate “rollback” to an evaluated configuration of a TOE if a future release is found to be flawed.

In order to effectively meet these two goals, it is necessary for the developer to show that the CM plan identifies appropriate CM items of the TOE (ACM_SCP.1) and that an appropriate CM plan is being followed (ACM_CAP.2).

Delivery and Operation

The inclusion of ADO_IGS.1 ensures that end users of a compliant TOE have appropriate guidance to generate, install and start-up a secure TOE.

Development

Because it is important to gain confidence in the design of the TOE, ADV_FSP.1, ADV_HLD.2 and ADV_RCR.1 are included to require the vendor to offer documentation-based support of evaluator analysis of the design. Specifically required is a functional specification (ADV_FSP.1) which describes the general behavior of the TOE, a high level design document (ADV_HLD.2) which describes the internal subsystems of the TOE and a mapping (ADV_RCR.1) that shows that these documents are consistent.

This documentation helps to satisfy all of the functional security objectives in that they require the documentation (and hence TOE understanding) necessary to determine whether a TOE function exists and is designed properly.

Guidance

The guidance documents class components, AGD_ADM.1 and AGD_USR.1, are included to ensure that proper TOE documentation is provided to end-users which explains the secure usage and administration of the TOE.

Tests

The test class components, ATE_COV.2, ATE_DPT.2, ATE_FUN.1, and ATE_IND.2 contribute to the overall confidence that the security functions have been implemented as designed. Specifically ATE_COV.2 mandates that analysis be done to show that all of the security functions have been tested. ATE_DPT.2 requires that the functional specification and the high level design be considered in test formulation. ATE_FUN.1 mandates the functional testing be performed by the developer and that the results be made available to the evaluator for review. ATE_IND.2 requests the evaluator to perform independent testing.

Vulnerability Assessment

The vulnerability class components require that the developer and the evaluator to perform a general search for vulnerabilities introduced by ambiguous documentation (AVA_MSU.1), probabilistic mechanisms (AVA_SOF.1) and TOE implementation flaws (AVA_VLA.1).

Development Security

The Development Security component (ALC_DVS.1) provides end users information about the security of the TOE development environment.

6.2.4.2 Assurance Requirements and Application Notes

This section presents a listing of each assurance component and its application notes. These notes provide an explanation of terminology and expected interpretations of meanings, and should aid in the analysis of requirement satisfaction for users of the profile.

ACM_CAP.2: Authorization Controls

ACM_CAP.2.1D**:** The developer shall use a CM system.

ACM_CAP.2.2D**:** The developer shall provide CM documentation.

ACM_CAP.2.1C**:** The CM documentation shall include a configuration list and a CM plan.

ACM_CAP.2.2C**:** The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.2.3C**:** The CM documentation shall describe the method used to uniquely identify the TOE configuration items.

ACM_CAP.2.4C**:** The CM plan shall describe how the CM system is used.

ACM_CAP.2.5C**:** The CM documentation shall provide evidence that the CM system is working properly.

ACM_CAP.2.6C**:** The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.

ACM_CAP.2.7C**:** The CM system shall ensure that only authorized changes are made to the TOE configuration items.

ACM_CAP.2.1E**:** The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Application Notes

The developer is required to provide a list of configuration items (CIs) that comprise the TOE and the developer's methodology that ensures that configuration items are uniquely identified in the configuration management (CM) system. The component ACM_SCP.1 identifies the aspects of the TOE and its development that shall be depicted as CIs.

The developer is also required to provide a CM plan, which describes how the CM system is used and should include the following:

a) How the CM system is used, including who and under what circumstances they are allowed to make modifications to a CI. There should be a description of how the CM system ensures that only authorized individuals are allowed to make changes to items under the control of the CM system.

b) The policy for using emergency procedures for correcting errors and for incorporating these corrections in subsequent scheduled product releases.

c) The procedures the developer uses for performing an in-house audit of the CM process.

d) A rationale for the chosen granularity of CIs.

e) A description of the format of evidence that the CM system is working properly and that all CIs have been and are being effectively maintained under the CM system.

f) The procedures for CM plan maintenance.

g) All updates necessary to reflect corrective measures taken after a CM process failure (e.g., failure to follow, or error in following, the CM plan), if one has occurred.

h) Evidence that the CM system is effectively controlling modification to CIs. This evidence should be a sampling of changes that have been made to CIs while the CIs were under the control of the CM system. The evidence should include the following:

   1) A description of the change made to the CI

   2) Accountability for the change(s)

   3) Identification of the CIs affected

   4) Status (e.g., being implemented, or completed) of the changes to the CIs

   5) All other information about the change that is maintained by the product's CM system.

ACM_SCP.1: Minimal CM Coverage

ACM_SCP.1.1D: The developer shall provide CM documentation.

ACM_SCP.1.1C: As a minimum, the following shall be tracked by the CM system: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, and CM documentation.

ACM_SCP.1.2C: The CM documentation shall describe how configuration items are tracked by the CM system.

ACM_SCP.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Application Notes

The purpose of this requirement is to ensure that the CM system employed by the developer is tracking all the necessary configuration items that comprise the TOE. In order to satisfy this requirement, the identified CIs should encompass:

a) The components or subsystems that comprise the TOE. In a software-only TOE, the implementation representation may consist solely of source and object code, but in TOEs consisting of a hardware platform, the implementation representation should refer to a combination of software, hardware and firmware.

b) Any hardware and/or software features that are used to periodically validate the correct operation of the TSF in satisfaction of the FPT_AMT.1 component.

c) The documentation used to meet the ADV_FSP.1 (ADV_FSP.2 for B1) component.

d) The documentation used to meet the AGD_USR.1 component.

e) The documentation used to meet the AGD_ADM.1 component.

f) The test plan, the test procedures that show how the security mechanisms were tested, and the expected results of the security mechanisms' functional testing, and related test documentation. This should include all documentation used to meet the ATE_COV.2, ATE_DPT.2, and ATE_FUN.1 components.

g) The design documentation used to satisfy the ADV_HLD.2 component.

h) The CM documentation itself, including the CM plan used to satisfy the ACM_CAP.2 component.

Additionally, the CM documentation should describe how the CIs are tracked by the CM system. This should include some high-level description of the developer's development/ maintenance process and how they relate to the CM process. Some of the things discussed should be how CIs are assigned an identifier, when and how that identifier is entered into the CM system, and how that identifier can be used to follow a CI through the CM process. Other things to consider are:

a) Does the CM system keep one or more versions of the CI, if so, how is that managed?

b) Does the CM system track what stage (e.g., open - waiting for security analysis, closed, pending for funding) of development/maintenance a CI is in?

c) Are CIs associated to one another when appropriate? If a change is made to a CI (e.g., a command's parameter) is there an indication that another CI (e.g., testing, user documentation) requires a change?

The CM documentation the developer provides to meet this requirement may be combined with the CM documentation delivered for the ACM_CAP.2 documentation.

ADO_IGS.1: Installation, Generation, and Start-up Procedures

ADO_IGS.1.1D: The developer shall document procedures to be used for the secure installation, generation, and start-up of the TOE.

ADO_IGS.1.1C: The documentation shall describe the steps necessary for secure installation, generation, and start-up of the TOE.

ADO_IGS.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Application Notes

The generation requirements are applicable only to TOEs that provide the ability to generate an operational TOE from source or object code.

The installation, generation, and start-up procedures may exist as a separate document, but would typically be grouped with other administrative guidance. The vendor is expected to provide the document containing the procedures while the evaluator reviews the document to ensure that it conforms to the listed requirements.

“Installation” includes such tasks as attaching peripheral or auxiliary components (mouse, keyboard, CD-ROM drive, tape drive); connecting to an electrical outlet, network, or other device; installing printed-circuit boards, chips, and memory devices; enabling the operating-system software corresponding to the input/output devices installed; and, loading software and data into the storage areas of an operating system.

“Generation” includes the tasks of compiling and linking source code (as well as any required source-code macros, header files, and external code called by the software) into executable code.

“Configuration” includes such tasks as choosing the operating characteristics, functions, and optional features that will be enabled when the TOE is started up and operating; installing and populating the data structures required for operation; and identifying users, creating user IDs, and establishing TOE privileges.

“Start-up” involves starting a particular configuration or instance of the TOE and, if applicable, a particular operating mode. Starting up the TOE may include depressing device ON/OFF switch(es), executing a list of operating-system or TSF interfaces, or submitting a file of job-control-language statements for immediate or scheduled execution.

“Maintenance” includes such tasks as keeping software and data applicable to TOE operations correct, current, and complete; managing and scheduling the installation and implementation of new TOE hardware and software releases; monitoring and improving TOE performance; performing backup/recovery procedures; and adding, deleting, and changing users, user IDs, and privileges.

ADV_FSP.1: TOE and Security Policy

ADV_FSP.1.1D: The developer shall provide a functional specification.

ADV_FSP.1.2D: The developer shall provide a TSP.

ADV_FSP.1.1C: The functional specification shall describe the TSF using an informal style.

ADV_FSP.1.2C: The functional specification shall include an informal presentation of syntax and semantics of all external TSF interfaces.

ADV_FSP.1.3C: The functional specification shall include evidence that demonstrates that the TSF is completely represented.

ADV_FSP.1.4C: *The functional specification shall include the following information:*[12]

a) *[FAU_GEN.2]* A specification of the manner in which an auditable event is associated with the identity of a user.[13]

b) *[FIA_ADP.1]* A specification of the manner by which the user authentication data is protected from unauthorized use, observation, modification and destruction while the authentication data is stored in the TSF.

c) *[FIA_ADP.2]* A specification of the manner by which the user authentication data is protected from unauthorized use, observation, modification and destruction at all times while it is under TSF control.

d) *[FPT_TSA]* A specification of the security-relevant administrative functions in the TSF.

e) *[FIA_ADA]* A specification of the TSF authentication data administration mechanism.

f) *[FIA_ATD]* A specification of the user-related TSP attributes and the manner in which they are associated with the user.

g) *[FIA_UAU]* A specification of the TSF authentication mechanism(s).

h) *[FIA_UID]* A specification of the user identification function.

i) *[FIA_USB]* A specification of the manner in which user attributes are associated with subjects that the user owns.

[12] This is a new element not contained in the CC. A CCOR has been written and submitted to the FRG [PPTeam1.0]. This element is a result of the FRG's recommendation.

[13] A CCOR concerning this element is currently being reviewed. [PPTeam51]

ADV_FSP.1.5C: *[FDP_ACC.1] The functional specification must define the subset of operations and objects controlled by the policy, describe the intended use of these operations, and provide a detailed rationale for the scope of the subset .*[14]

ADV_FSP.1.6C: *[FDP_ACC.1] The functional specification should be used to provide evidence for the rationale that the objects are covered by the access control Security Function Policies (SFPs) and that there are no conflicts in the case of multiple SFPs (See footnote 14).*

ADV_FSP.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.1.2E: The evaluator shall determine that the functional specification is consistent with the TSP.

ADV_FSP.1.3E: The evaluator shall determine if the functional requirements in the ST are addressed by the representation of the TSF.

Application Notes

The developer must provide evidence that the TSF is completely represented by the functional specification. While a functional specification for the entire TOE would allow an evaluator to determine the TSF boundary, it is not necessary to require that specification when other evidence could be provided to demonstrate the TSF boundary.

The evaluator of the TOE is expected to make determinations regarding the functional requirements in the ST relevant to the functional specification. In the course of the functional specification evaluation there are essentially three types of evaluator determination: specific functional requirements are met and no further work (e.g., with a less abstract representation of the TSF) is necessary; specific functional requirements are violated and the TOE fails to meet its requirements; and specific functional requirements have not been addressed and further analysis (of another TSF representation) is necessary. Whenever more analysis is necessary, the evaluator is expected to carry that information forward to the analysis of other TSF representations. If requirements are not addressed after the analysis of the last provided TSF representation, this also represents a failure of the TOE evaluation. Note that this more comprehensive failure determination requirement is realized in the Representation Correspondence (ADV_RCR) family.

There should be a description of how access controls are applied to each object protected by the TSF and how revocation of access takes place (i.e., immediately or when there are no references to the object).

Not all commands, system calls, or instructions represent a TSF interface. Some commands and library calls refer only to programs and data structures that are outside the TSF in which case they are not considered TSF interfaces.

[14] A CCOR concerning this element is currently being reviewed [PPTeam1.1].

Some command and application program interfaces may overlap and not represent distinct TSF interfaces. Security-relevant TSF interfaces are those interfaces that:

- Change the security state of the product;
- Permit an object access or information flow that is regulated by the security policy;
- Are restricted to subjects with privilege or behave differently when executed by subjects with privilege; and
- Implement or support a security mechanism.

ADV_HLD.2: Security Enforcing High-level Design

ADV_HLD.2.1D: The developer shall provide the high-level design of the TSF.

ADV_HLD.2.1C: The presentation of the high-level design shall be informal.

ADV_HLD.2.2C: The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.2.3C: The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.2.4C: *The high-level design shall describe the external interfaces of the TSF.*[15]

ADV_HLD.2.5C: The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV_HLD.2.6C: The high-level design shall describe the separation of the TSF into TSP enforcing and other subsystems.

ADV_HLD.2.7C: *The high-level design shall include the following information:*[16]

a) *[FIA_UID]* A specification of the user identification function.

b) *[FPT_SEP]* A description of the architecture and design of the domain separation mechanism.

ADV_HLD.2.8C: *The high-level design shall describe how the correct operation of the security-relevant functions provided by the TSF's underlying abstract machine is demonstrated.*[17]

ADV_HLD.2.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_HLD.2.2E: The evaluator shall determine if the functional requirements in the ST are addressed by the representation of the TSF.

Application Notes

The developer is expected to describe the design of the TSF in terms of subsystems. The term "subsystem" is used here to express the idea of decomposing the TSF into a relatively small number of parts. While the developer is not required to actually have "subsystems,"

[15] This modification to the CC has been submitted as a CCOR [c2pp97.1].
[16] This is a new element not contained in the CC. A CCOR has been written and submitted to the FRG [PPTeam1.0]. This element is a result of the FRG's recommendation.
[17] A CCOR concerning this new element has been written [c2pp97.9].

the developer is expected to represent a similar level of decomposition. For example, a design may be similarly decomposed using “layers,” “domains,” or “servers.”

The evaluator of the TOE is expected to make determinations regarding the functional requirements in the ST relevant to the high-level design. In the course of the high-level design evaluation there are essentially three types of evaluator determination: specific functional requirements are met and no further work (e.g., with a less abstract representation of the TSF) is necessary; specific functional requirements are violated and the TOE fails to meet its requirements; and specific functional requirements have not been addressed and further analysis (of another TSF representation) is necessary. Whenever more analysis is necessary, the evaluator is expected to carry that information forward to the analysis of other TSF representations. If requirements are not addressed after the analysis of the last provided TSF representation, this also represents a failure of the TOE evaluation. Note that this more comprehensive failure determination requirement is realized in the Representation Correspondence (ADV_RCR) family.

The term “security functionality” is used to represent operations that a subsystem performs that have some effect on the security functions implemented by the TOE. This distinction is made because design constructs, such as subsystems and modules, do not necessarily relate to specific security functions. While a given subsystem may correspond directly to a security function, or even multiple security functions, it is also possible that many subsystems must be combined to implement a single security function.

The term “TSP enforcing subsystems” refers to a subsystem that contributes to the enforcement of the TSP.

With respect to the identification of the user, a description of what constitutes a successful logon, the logon process, access control checks performed after successful authentication, and what constitutes a logon failure should be present.

The expected constraints and outcomes of the security-relevant TSF operations should be documented in such that an argument is given as to why the operations preserve the secure state of the TOE.

There should be a description of the security features of the TOE that allow for verification of the correct functioning of the hardware and firmware upon which the TSF relies to enforce the security policy.

ADV_RCR.1: Informal Correspondence Demonstration

ADV_RCR.1.1D: The developer shall provide evidence that the least abstract TSF representation provided is an accurate, consistent, and complete instantiation of the functional requirements expressed in the ST.

ADV_RCR.1.1C: For each adjacent pair of TSF representations, the evidence shall demonstrate that all parts of the more abstract representation are refined in the less abstract representation.

ADV_RCR.1.2C: For each adjacent pair of TSF representations, the demonstration of correspondence between the representations may be informal.

ADV_RCR.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_RCR.1.2E: The evaluator shall analyze the correspondence between the functional requirements expressed in the ST and the least abstract representation provided to ensure accuracy, consistency, and completeness.

Application Notes

It is up to the developer to provide the necessary analysis for the evaluator. This analysis should demonstrate the least abstract representation of the TSF is consistent with the selected functional components.

It is up to the evaluator to determine whether the developer’s analysis is accurate, complete, and consistent. This is done by examining each demonstration of correspondence between abstractions and the results of the developer’s analysis. The evaluator should then determine whether all the functional requirements have been satisfied.

This family of requirements is not intended to address correspondence relating to the TSP model or the TSP. Rather, as shown in Figure 5.4 of the Common Criteria 1.0, it is intended to address correspondence between the requirements in the ST as well as the TOE summary specification, functional specification, high-level design, low-level design, and implementation representation.

AGD_ADM.1: Administrator Guidance

AGD_ADM.1.1D: The developer shall provide administrator guidance addressed to system administrative personnel.

AGD_ADM.1.1C: The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD_ADM.1.2C: The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_ADM.1.3C: The administrator guidance shall contain guidelines on the consistent and effective use of the security functions within the TSF.

AGD_ADM.1.4C: The administrator guidance shall describe the difference between two types of functions: those which allow an administrator to control security parameters, and those which allow the administrator to obtain information only.

AGD_ADM.1.5C: The administrator guidance shall describe all security parameters under the administrator's control.

AGD_ADM.1.6C: The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_ADM.1.7C: The administrator guidance shall contain guidelines on how the security functions interact.

AGD_ADM.1.8C: The administrator guidance shall contain instructions regarding how to configure the TOE.

AGD_ADM.1.9C: The administrator guidance shall describe all configuration options that may be used during secure installation of the TOE.

AGD_ADM.1.10C: The administrator guidance shall describe details, sufficient for use, of procedures relevant to the administration of security.

AGD_ADM.1.11C: The administrator guidance shall be consistent with all other documents supplied for evaluation.

AGD_ADM.1.12C: *[FAU_GEN] The administrator guidance shall provide a description of the audit record format.*[18]

[18] A CCOR has been submitted to the FRG [PPTeam1.0] and an FRG response has been forwarded to the team. However, the recommendation was to create a new element, AGD_ADM.1.12C which is currently AGD_ADM.1.14C in this profile**.** A separate CCOR has been written to address the use of "should" and "shall" within the documentation notes [PPTeam1.1]. This element was introduced to the profile as a result of pulling a documentation note that used "should" and making it a required element.

AGD_ADM.1.13C: *[FAU_GEN] The administrator guidance shall provide a list of auditable events* [19].

AGD_ADM.1.14C: *The administrator guidance shall contain guidance on or a description of:* [20]

a) *[FAU_MGT]* Recommendations for handling notifications generated by the TSF when a security violation appears imminent[21].

b) *[FAU_MGT]* Recommendations for defining the limit to control the audit trail saturation[22].

c) *[FAU_SAR]* Using the audit review tools.[23]

d) *[FDP_ACI.1]* Identify acceptable alternate initial values for object security attributes if the default values are over-ridden.[24]

e) *[FDP_SAM.1]* How security attributes associated with objects, users, or subjects are modified.[25]

f) *[FIA_ADA]* Using the TSF authentication data administration mechanism.

g) *[FIA_UAU]* Configuring the TSF authentication mechanism(s).

h) *[FIA_UAU.8]* Explaining the risks in placing passwords on card input and suggesting procedures to mitigate that risk.

i) *[FIA_UID]* How to define users.

j) *[FPT_AMT]* Using the product features that can be used to periodically demonstrate the correct operation of the underlying abstract machine

k) *[FPT_AMT]* The coverage and use of the underlying abstract machine tests.

l) *[FPT_REV.1]* The timing aspects of the revocation.

m) *[FPT_TSA]* The initial configuration of the security-relevant administrative commands and (if applicable) the roles with which they are associated.

[19] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.
[20] This is a new element not contained in the CC. A CCOR has been written and submitted to the FRG [PPTeam1.0]. This element is a result of the FRG's recommendation.
[21] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.
[22] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.
[23] A CCOR concerning this element is currently being reviewed. [PPTeam50]
[24] A CCOR concerning this element is currently being reviewed. [PPTeam52]
[25] A CCOR concerning this element is currently being reviewed. [PPTeam53]

n) *[FPT_TSA]* The TSF facilities used by an authorized administrator to define security-relevant administrative commands and (if applicable) associate them with a role.

o) *[FPT_TSA]* The responsibilities of the security-relevant administrative role(s), as applicable.

p) *[FAU_MGT]* Recommendations for handling notifications generated by the TSF when a security violation appears imminent[26].

q) *[FAU_MGT]* Recommendations for defining the limit to control the audit trail saturation[27].

r) *[FAU_PRO]* A description of the protection rules for the audit trail *(See footnote 14)*.

s) *[FAU_PRO]* An identification of the rules for managing access to the audit trail by users[28].

t) *[FAU_SEL]* A description of the selection rules for the audit events *(See footnote 14)*.

u) *[FAU_SEL]* An identification of the rules for managing the auditable set of events[29].

v) *[FAU_STG]* An identification of the conditions under which loss of audit data due to system failure shall be enumerated and the potential number of audit events lost shall be documented[30].

w) *[FDP_ACC]* Guidance with respect to each access control policy satisfying a FDP_ACC component. Documentation shall be provided for end-users, authorized administrative users, or both, as appropriate for the nature of the objects and operations controlled by the policy[31].

x) *[FDP_ACC.1]* A definition of the subset of operations controlled by the SFP, describe the intended use of the operations, and provide a detailed rationale for the scope of the subset. The rationale shall be sufficient to convince the evaluator that all listed objects are covered by the access control SFP. In the case of multiple SFPs, rationale should be provided to demonstrate that the SFPs do not conflict. If the PP/ ST author claims that there is complete coverage of all objects and operations within the TOE Scope of Control (TSC), then rationale shall be provided to demonstrate this as well as that no conflicts exist between the SFPs[32].

[26] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.
[27] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.
[28] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.
[29] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.
[30] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.
[31] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.

y) *[FDP_ACF]* Information detailing what is the basis of mediation, what is the precedence of mediation when more than one conclusion could be reached given a set of attributes, etc.[33].

z) *[FDP_ACF]* A description of the nature and scope of each access control policy and briefly describe the functions that implement the policy (FDP_ACF), the security attributes that govern the policy (FDP_SAQ, FDP_SAM), the initialisation rules for those attributes (FDP_ACI), and (if any) the default mechanisms for those attributes (FDP_ACI)[34].

aa) *[FDP_ACF]* Guidance on the safe and effective use of the mechanisms[35].

ab) *[FAU_STG]* A list the conditions under which loss of audit data due to system failure shall be enumerated and the potential number of audit events lost should be documented[36].

AGD_ADM.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_ADM.1.2E: The evaluator shall confirm that the installation procedures result in a secure configuration.

Application Notes

The administrator guidance documentation should be provided by the developer while the evaluator should review the documentation to determine whether it complies with the requirements listed above.

The administrator guidance should define the operations controlled by the policy, describe the intended use of the operations, and provide a detailed rationale for the scope of the subset. The rationale should be sufficient to convince the evaluator that all listed objects and operations are covered by the access control Security Function Policy (SFP).

[32] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.
[33] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.
[34] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.
[35] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.
[36] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.

AGD_USR.1: User Guidance

AGD_USR.1.1D: The developer shall provide user guidance.

AGD_USR.1.1C: The user guidance shall describe the TSF and interfaces available to the user.

AGD_USR.1.2C: The user guidance shall contain guidelines on the use of security functions provided by the TOE.

AGD_USR.1.3C: The user guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4C: The user guidance shall describe the interaction between user-visible security functions.

AGD_USR.1.5C: The user guidance shall be consistent with all other documentation delivered for evaluation.

AGD_USR.1.6C: *The user guidance shall provide guidance on:*

a) *[FIA_UAU]* Use of the TSF authentication mechanism(s).

b) *[FIA_UID]* How to identify themselves to the TOE.

c) *[FDP_ACC.1]* Guidance with respect to each access control policy satisfying a FDP_ACC component. Documentation shall be provided for end-users, authorized administrative users, or both, as appropriate for the nature of the objects and operations controlled by the policy[37].

d) *[FDP_ACF]* Information detailing what is the basis of mediation, what is the precedence of mediation when more than one conclusion could be reached given a set of attributes, etc.[38].

e) *[FDP_SAM.2]* How security attributes associated with objects, users, or subjects are modified.[39]

AGD_USR.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Application Notes

No Application Notes are provided for this component.

[37] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.
[38] A CCOR [PPTeam1.1] was submitted to FRG. Received FRG agreement.
[39] A CCOR concerning this element is currently being reviewed. [PPTeam54]

ALC_DVS.1: Identification of Security Measures

ALC_DVS.1.1D: The developer shall produce development security documentation.

ALC_DVS.1.1C: The development security documentation shall describe the physical, procedural, personnel, and other security measures that are used to protect the confidentiality and integrity of the TOE during its development.

ALC_DVS.1.2C: The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

ALC_DVS.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DVS.1.2E: The evaluator shall check whether the security measures are being applied.

Application Notes

Developer actions are performed by the vendor and evaluator actions are performed by an evaluator.

The evaluator should decide whether there is a need for visiting the user’s site in order to confirm that the requirements of this family are met.

The evaluation scheme should provide for a way to make the development security documentation, or a summary of it, available to system users such as certifiers, integrators, and purchasers.

ATE_COV.2: Complete Coverage - Rigorous

ATE_COV.2.1D: The developer shall provide an analysis of the test coverage.

ATE_COV.2.1C: The analysis of the test coverage shall demonstrate that the tests identified in the test documentation cover the TSF.

ATE_COV.2.2C: The analysis of the test coverage shall demonstrate the correspondence between the security functions and the tests identified in the test documentation.

ATE_COV.2.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Application Notes

The developer is required to provide an analysis of test coverage that demonstrates that all the security functions are tested. The analysis to be provided can be informal and may take one of many forms (e.g., a matrix) as long as it provides a mapping between security functions and test cases. The important aspect is that an evaluator can determine that the developer's test suite covers all the security functions. In order to ensure complete coverage, the developer should have identified all of the TSF external interfaces, as well as identifying the security functions that each interface supports. The developer then provides a mapping of test cases that provide test coverage.

For this requirement it is expected that the security-relevant aspects of every external TSF interface has one or more tests that demonstrates that the interface exhibits the proper behavior. An interface may support more than one security policy (e.g., access control and audit) and the test suite must address each policy. An interface may also incorporate more than one aspect of a given policy (e.g., the interface may operate on different object types and will behave differently depending on the object type) and each of those aspects requires testing as well.

ATE_DPT.2: Testing - High Level Design

ATE_DPT.2.1D: The developer shall provide the analysis of the depth of testing.

ATE_DPT.2.1C: The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TOE operates in accordance with the functional specification, and high level design of the TSF.

ATE_DPT.2.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Application Notes

The evaluator is responsible for ensuring that the developer's test suite provides enough depth of coverage to demonstrate that the TOE operates in accordance with the functional specification and the high-level design of the TSF. Whereas the evaluator ensures breadth of coverage in ATE_COV, here the evaluator must ensure that interfaces are tested in sufficient detail to ensure that they are correctly implemented with respect to the security policies. This requires the evaluator to examine the level of testing performed on the external interfaces, which the functional specification represents, and the internal “subsystem” interfaces, which are described in the high-level design. At this level of assurance, the evaluator is not expected to examine 100% of the interfaces, but rather a representative sample of interfaces. Typically an evaluator will examine 50% - 60% of the interfaces and ensure that all subsystems are covered. If problems are discovered the sample size may be increased depending on the extent of the problems found.

Depth of testing is related to both the interface level (external for ATE_DPT.1, external and a degree of internal interfaces for ATE_DPT.2) of testing and the degree in which an interface is tested. When determining the degree to which an interface is tested, depth involves the extent to which both normal and abnormal uses of an interface are tested. With respect to normal use, depth involves testing multiple combinations of values of input parameters to interfaces, rather than testing a few invocations that are deemed representative of expected use. With respect to abnormal uses, the evaluator ensures that the vendor has tested boundary conditions in the use of interfaces, as well as expected uses. Exhaustive or extensive boundary testing is not required; however, the evaluator ensures that the vendor tests against stated warnings in the documentation and that simple boundary conditions (small deviations from boundary values, such as might result from user errors) have been covered.

Test coverage depth depends strongly on (1) the security policy or policies the vendor claims the TOE enforces and (2) the variety of actions that invoke policy enforcement mechanisms and the number of parameters associated with those actions. Sufficiently deep testing explores interactions between policies or between mechanisms intended to enforce the same policy. Sufficiently deep testing addresses more than a representative sample of actions and parameter values. However, exhaustive testing of all possible combinations of parameter values and actions is impossible, even when automatic test generation tools are used.

ATE_FUN.1: Functional Testing

ATE_FUN.1.1D: The developer shall test the TSF and document the results.

ATE_FUN.1.2D: The developer shall provide test documentation.

ATE_FUN.1.1C: The test documentation shall consist of test plans, test procedure descriptions, and test results.

ATE_FUN.1.2C: The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE_FUN.1.3C: The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function.

ATE_FUN.1.4C: The test results in the test documentation shall show the expected results of each test.

ATE_FUN.1.5C: The test results from the developer execution of the tests shall demonstrate that each security function operates as specified.

ATE_FUN.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Application Notes

The developer is required to provide the evaluator test documentation, which includes the vendor's test plan, test procedures, test descriptions, and expected results. The vendor's test plan should discuss the developer's philosophy of testing the security functionality of the TOE. The test plan should address things such as:

- What type of tools, if any, does the developer rely on for testing?
- Does the developer intend on using gray-box testing?
- How does the developer intend on testing the residual information protection requirement?
- How does the developer test domain separation?
- The test plan should also discuss the developer's approach to testing. Do they have a quality assurance group that performs testing? Are the programmers responsible for testing their components?

For each test case the documentation should describe the test's purpose, a description of the test scenario (which should match the test's purpose) and the expected results of the test case. Expected results should be derived from design and interface documentation. A developers test suite may generate messages such as “TEST PASSES.” If this is the case, the test documentation must describe how these messages are generated and the evaluator must examine the documentation to ensure the messages are generated under the correct circumstances.

ATE_IND.2: Independent Testing - Sample

ATE_IND.2.1D: The developer shall provide the TOE for testing.

ATE_IND.2.1C: The TOE shall be suitable for testing.

ATE_IND.2.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2E: The evaluator shall test the TSF to confirm that the TSF operates as specified.

ATE_IND.2.3E: The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

Application Notes

Independent testing is performed by the evaluator to ensure the developer's test procedures are clear and repeatable, as well as to gain assurance that the developer's test suite runs as expected on the TOE in its evaluated configuration. As part of the independent testing the evaluator is expected to bring up the TOE in its evaluated configuration per the instructions provided by the administrator documentation (ADO_IGS.1). If the developer's TOE includes a wide range of hardware (e.g., single and multiple CPUs) the evaluator should consider running the tests on a representative sample of the hardware configurations.

Once the TOE is up and in its evaluated configuration the evaluator runs a representative sample of the developer's test suite. The sample of tests run should include sufficient breadth and depth of the developer's test suite to ensure that the sample is indeed representative of the test suite. If the evaluator experiences errors or unexpected results when running the tests the developer must fix the bug(s) in the TOE. The evaluator must then determine what tests must be rerun. It is possible that only the test that uncovered the bug needs to be rerun. In some cases the entire test suite may have to be rerun. If there are a number of errors uncovered, the evaluator should consider increasing the sample of tests to be executed.

While analyzing the test results, the evaluator should compare their test results with the expected results identified in the test documentation. Additionally, the evaluator should compare a sample of developer provided test results with those generated from the evaluator's independent execution of tests to ensure the results are consistent with one another, as well as with the expected results. Additionally, the evaluator should examine a sample of the developer's test results of tests that were not run by the evaluator to ensure they were consistent with the expected results.

AVA_MSU.1: Misuse Analysis - Obvious Flaws

AVA_MSU.1.1D: The developer shall document an analysis of the guidance documentation for conflicting and incomplete guidance.

AVA_MSU.1.2D: The developer shall ensure that the guidance documentation contains no misleading or unreasonable guidance.

AVA_MSU.1.1C: The analysis documentation shall provide a rationale that demonstrates that the guidance is not conflicting and is complete.

AVA_MSU.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_MSU.1.2E: The evaluator shall determine that there is no misleading or unreasonable guidance in the guidance documentation.

AVA_MSU.1.3E: The evaluator shall repeat any procedures in the guidance documentation to ensure that they produce the documented results.

Application Notes

The intent of the developer actions of this requirement is to mandate that the developer explicitly review the TOE guidance documentation specifically for inconsistent or incomplete guidance. Extensive or unusual documentation of the results of this analysis is not required to meet AVA_MSU.1.1D

An example of conflicting guidance would be two guidance instructions which imply different outcomes when the same input is supplied.

An example of misleading guidance would be the description of a single guidance instruction which could be parsed in more than one way, one of which may result in an insecure state.

An example of completeness would be referencing assertions of dependencies on external security measures e.g., such as external procedural, physical and personnel controls.

AVA_SOF.1: Strength of TOE Security Function Evaluation

AVA_SOF.1.1D: The developer shall identify all TOE security mechanisms for which a strength of TOE security function analysis is appropriate.

AVA_SOF.1.2D: The developer shall perform a strength of TOE security function analysis for each identified mechanism.

AVA_SOF.1.1C: The strength of TOE security function analysis shall determine the impact of the identified TOE security mechanisms on the ability of the TOE security functions to counter the threats.

AVA_SOF.1.2C: The strength of TOE security function analysis shall demonstrate that the identified strength of the security functions is consistent with the security objectives of the TOE.

AVA_SOF.1.3C: Each strength claim shall be either basic, medium, or high.

AVA_SOF.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_SOF.1.2E: The evaluator shall confirm that all TOE security mechanisms requiring a strength analysis have been identified.

AVA_SOF.1.3E: The evaluator shall confirm that the strength claims are correct.

Application Notes

The vendor is responsible for producing the documentation containing the initial analysis while the evaluator is responsible for reviewing it.

The evaluator confirms the developer's assessment by examining the developer's security function analysis.

The evaluator examines the information provided by the developer in AVA_SOF.1.2D and contrasts this information against the interpretations and experience of previous TOE evaluations. If the mechanism employed is one for which there is no existing interpretation or other evaluator body of evidence, the evaluator may employ other means to validate the strength claims. Other means include requesting further detail from the developer (via the sponsor) on the capabilities and limitations of the mechanism, and checking with external organizations (external to evaluation scheme) to see if they have experience with the particular mechanism.

In those instances where a single security mechanism is employed for enforcing multiple security functions, the evaluator confirms that the developer has performed a strength analysis on the mechanism for each security function that the mechanism enforces. Alternatively, where multiple security mechanisms are required to enforce a single security function, the strength analysis should note the dependence of multiple mechanisms on each other.

The evaluator should also confirm that the developer's security function analysis references relevant assertions in documentation developed in support of other assurance components.

The evaluator should verify that the developer has identified all of the security policies noted in the TSP.

This component applies only to those functions that involve a measurable risk that often can be expressed probabilistically. This component does not apply to those functions that are absolute in their security policy enforcement.

Security functions are implemented by security mechanisms. For example, a password mechanism can be used in the implementation of the identification and authentication security function.

The strength of TOE security functions evaluation is performed at the level of the security mechanism, but its results provide knowledge about the ability of the related security function to counter the identified threats.

The strength of a function is rated "basic" if the analysis shows that the function provides adequate protection against unintended or casual breach of TOE security by attackers possessing a low attack potential.

The strength of a function is rated "medium" if the analysis shows that the function provides adequate protection against attackers possessing a moderate attack potential.

The strength of a function is rated "high" if the analysis shows that the function provides adequate protection against attackers possessing a high attack potential.

The attack potential is derived from the attacker's expertise, opportunities, resources, and motivation.

AVA_VLA.1: Developer Vulnerability Analysis

AVA_VLA.1.1D: The developer shall perform and document an analysis of the TOE deliverables searching for obvious ways in which a user can violate the TSP.

AVA_VLA.1.2D: The developer shall document the disposition of identified vulnerabilities.

AVA_VLA.1.1C: The evidence shall show, for each vulnerability, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA_VLA.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VLA.1.2E: The evaluator shall conduct penetration testing, based on, but not limited to[40], the developer vulnerability analysis, to ensure obvious vulnerabilities have been addressed.

Application Notes

The evaluator should consider the following with respect to the search for obvious flaws:

a) Dependencies among functional components and potential inconsistencies in strength of function among interdependent functions (*vide* ASE_REQ.1.10C and AVA_SOF.1);

b) Potential inconsistencies between the TSP and the functional specification (*vide* ADV_FSP.1.2E);

c) Potential gaps or inconsistencies in the HLD, and potentially invalid assumptions about supporting hardware, firmware, and/or software required by the TSF (*vide* ADV_HLD.2);

d) Potential gaps in the administrator guidance that enable the administrator to fail (a) to make effective use of TSF functions, (b) to understand or take actions that need to be performed, (c) to avoid unintended interactions among security functions, and (d) to install and/or configure the TOE correctly. In particular, failure to describe all the security parameters under the administrator's control and the effects of settings of (interacting combinations of) those parameters (*vide* AGD_ADM.1);

e) Potential gaps in the user guidance that enable the user to fail to control functions and privileges as required to maintain a secure processing environment. Potential presence in the user guidance of information that facilitates exploitation of vulnerabilities (*vide* AGD_USR.1).

[40] A CCOR has been submitted for this modification to the CC [c2pp97.2].

f) Open literature (e.g., CERT advisories, bug-traq mailing list) may contain information on vulnerabilities on the TSF and these sources should be consulted.

The evaluator confirms that the developer has addressed, for each vulnerability found, whether the vulnerability can be exploited by actions taken at the user interface or by processes acting on behalf of users at the TSF interface. (This activity depends on AGD_USR.1; the developer must have described TSF interfaces available to the user and the evaluator must have confirmed the completeness and accuracy of this description.)

6.2.5 Requirements Dependency Analysis

This section presents an analysis of component dependencies. Functional components possess dependencies which are stated requirements for the CAP PP to include further components in support of the primary requirements. To meet the evaluation requirements, it is necessary for all dependencies to be satisfied. Table 6.3 below demonstrates how the dependencies of each included component have been satisfied.

All the components of the CAP PP are listed with a numeric reference (Ref#). The dependencies of each component are listed alongside that component with the reference of that component within the table (see “Coverage” column). In the “Status” column, each component receives a “---” or a “NOT SATISFIED” as a rating. A “NOT SATISFIED” status means that the dependency is not included in the profile. A “---” means that the dependency is satisfied. In some cases a reason is given for the dependency. If the dependency is not satisfied, the reason is a reference to a “Remark,” which provides an explanation as to why the dependency is not satisfied. The Remarks follow the table. If the dependency is satisfied by a component that is hierarchically above the component listed as the dependency, “hierarchy” is listed as the reason. The table demonstrates that the CAP PP has satisfied all dependencies with respect to the chosen components with the few exceptions noted below.

Ref#	Component	Dependencies	Coverage	Status	Reason
FUNCTIONAL COMPONENTS					
1	FAU_GEN.1	FIA_UID.1	---	**NOT SATISFIED**	see Remark C
2	FAU_GEN.2	FAU_GEN.1 FIA_UID.1	1 ---	--- **NOT SATISFIED**	--- see Remark C
3	FAU_MGT.1	FAU_STG.1	9	---	hierarchy
4	FAU_MGT.2	FAU_STG.1	9	---	hierarchy
5	FAU_PRO.2	FAU_STG.1 FPT_TSA.1	9 29	--- ---	hierarchy ---
6	FAU_SAR.2	FAU_PRO.2 FAU_SAR.1 FAU_STG.1 FPT_TSA.1	5 9 29	--- **NOT SATISFIED** --- ---	hierarchy see Remark E hierarchy ---
7	FAU_SAR.3	FAU_PRO.2 FAU_SAR.1 FAU_STG.1 FPT_TSA.1	5 --- 9 29	--- **NOT SATISFIED** --- ---	hierarchy see Remark E hierarchy ---
8	FAU_SEL.1	FAU_GEN.1	1	---	---
9	FAU_STG.3	FAU_GEN.1	1	---	---
10	FDP_ACC.1	FDP_ACF.1	11	---	---
11	FDP_ACF.1	FDP_ACC.1	10	---	---
12	FDP_ACI.1	FDP_ACC.1	10	---	---
13	FDP_RIP.3	---	---	---	---

Table 6.3 - Component Dependency Analysis (Functional and Assurance)

14	FDP_SAM.2	FDP_ACC.1 FPT_TSA.1	10 29	--- ---	--- ---
15	FIA_ADA.3	FIA_ADP.1 FIA_UAU.1 FPT_TSA.1	16 --- 29	--- **NOT SATISFIED** ---	--- see Remark A ---
16	FIA_ADP.1	FIA_UAU.1	---	**NOT SATISFIED**	see Remark A
17	FIA_ADP.2	FIA_UAU.1	---	**NOT SATISFIED**	see Remark A
18	FIA_ATA.3	FIA_ATD.1 FPT_TSA.1	19 29	---	---
19	FIA_ATD.1	ADV_FSP.1	33	---	---
20	FIA_UAU.8	FIA_UAU.1	---	**NOT SATISFIED**	see Remark A
21	FIA_UID.2	---	---	---	---
22	FIA_UID.3	---	---	---	---
23	FIA_USB.1	FDP_ACI.1 FIA_ATD.1 ADV_FSP.1	12 19 33	--- --- ---	--- hierarchy ---
24	FPT_AMT.1	---	---	---	---
25	FPT_REV.1	---	---	---	---
26	FPT_REV.2	---	---	---	---
27	FPT_RVM.1	---	---	---	---
28	FPT_SEP.1	---	---	---	---
29	FPT_TSA.1	FIA_ATA.1 FIA_ATD.1 FIA_UID.1 AGD_ADM.1	--- 19 21 36	--- --- **NOT SATISFIED** ---	 hierarchy see Remark C ---
ASSURANCE COMPONENTS					
30	ACM_CAP.2	ACM_SCP.1 ALC_DVS.1	31 38	--- ---	--- ---
31	ACM_SCP.1	ACM_CAP.2	30	---	---
32	ADO_IGS.1	AGD_ADM.1	36	---	---
33	ADV_FSP.1	ADV_RCR.1 ASE_TSS.1	35 ---	--- **NOT SATISFIED**	--- see Remark D
34	ADV_HLD.2	ADV_FSP.1 ADV_RCR.1	33 35	--- ---	--- ---
35	ADV_RCR.1	---	---	---	---
36	AGD_ADM.1	ADV_FSP.1	33	---	---
37	AGD_USR.1	ADV_FSP.1	33	---	---
38	ALC_DVS.1	---	---	---	---
39	ATE_COV.2	ADV_FSP.1 ATE_FUN.1	33 40	--- ---	--- ---
40	ATE_DPT.2	ADV_FSP.1 ATE_FUN.1	33 40	--- ---	--- ---

Table 6.3 - Component Dependency Analysis (Functional and Assurance)

40	ATE_FUN.1	ATE_COV.1 ATE_DPT.1	38 39	--- ---	--- ---
41	ATE_IND.2	ADV_FSP.1 AGD_ADM.1 AGD_USR.1 ATE_FUN.1	32 35 36 40	--- --- --- ---	--- --- --- ---
42	AVA_MSU.1	ADO_IGS.1 AGD_ADM.1 AGD_USR.1	31 35 36	--- --- ---	--- --- ---
43	AVA_SOF.1	ADV_FSP.1 ADV_HLD.1	32 33	--- ---	--- hierarchy
44	AVA_VLA.1	ADV_FSP.1 ADV_HLD.1 AGD_ADM.1 AGD_USR.1	32 33 35 36	--- --- --- ---	--- hierarchy --- ---

Table 6.3 - Component Dependency Analysis (Functional and Assurance)

The dependencies of the requirements components of the TOE are satisfied by the inclusion of the relevant component within the TOE security requirements listed in this protection profile with the following exceptions:

Remark A. FIA_ADA.3, FIA_ADP.1, FIA_ADP.3, and FIA_UAU.8 lists FIA_UAU.1 as a dependency. This appears to be a broken dependency. FIA_UAU.8 requires that users be allowed to perform a set of functions prior to authentication while FIA_UAU.1 requires that no functions be performed prior to authentication. Both elements cannot exist simultaneously as they contradict each other.

Remark B. FIA_TSA.1 lists FIA_ATA.1 as a dependency. FIA_ATA.1 requires that the TOE produce default values for user attributes rather than place the burden on the administrator by describing a process in the administrator guidance documentation. While the requirement may be useful, it does not map to any of the security objectives for this protection profile and it appears that adequate manual procedures with documentation support could cover the intent of this component. This dependency may, however, be useful in other protection profiles (i.e., other security environments) and it is therefore not recommended that this dependency be removed entirely from the Common Criteria. Rather, this dependency should be waived for this protection profile.

Remark C. FIA_UID.3 replaces the functionality of FIA_UID.1.

Remark D. ADV_FSP lists ASE_TSS.1 as a dependency. This component is part of the Security Target Evaluation class which is beyond the scope of this profile and, therefore, is not included. This dependency should not exist within any of the functional or assurance components, as it is a requirement for a Security Target and excludes the notion that the component (i.e., ADV_FSP) would only be used in a Security Target specification. In addition, the ASE_TSS.1 component lists dependencies that may, in turn, list more dependencies. The concern is that many, if not all, of those indirect dependencies would be irrelevant to this protection profile.

Remark E. FIA_SAR.3 lists FAU_SAR.1 as a dependency. However, since FAU_SAR.2 is included in this profile and is hierarchical to FAU_SAR.1, the dependency is considered to be implicitly met.